UNITED STATES DISTRICT COURT
IN THE WESTERN DISTRICT OF TEXAS
WACO DIVISION

DATA SCAPE LIMITED, Plaintiff, v. TERADATA OPERATIONS, INC., Defendant.	C.A. No. 6:19-cv-00309 **JURY TRIAL DEMANDED**

COMPLAINT FOR PATENT INFRINGEMENT

This is an action for patent infringement arising under the Patent Laws of the United States of America, 35 U.S.C. § 1 *et seq.* in which plaintiff Data Scape Limited ("Plaintiff," "Data Scape") makes the following allegations against defendant Teradata Operations, Inc. ("Defendant"):

PARTIES

1. Data Scape is a company organized under the laws of Ireland with its office located at Office 115, 4-5 Burton Hall Road, Sandyford, Dublin 18, Ireland.

2. On information and belief, Defendant is Defendant Teradata Operations, Inc. ("Teradata" or "Defendant") is a Delaware corporation with its principal office at 10000 Innovation Drive, Dayton, Ohio 45342. On information and belief, Teradata maintains offices within this District at 9390 Research Boulevard, 3rd Floor, Austin, TX 78759. On information and belief, Teradata can be served through its registered agent, C T Corporation System, 818 West Seventh St Suite 930, Los Angeles, CA 90017.

JURISDICTION AND VENUE

3. This action arises under the patent laws of the United States, Title 35 of the United States Code. This Court has original subject matter jurisdiction pursuant to 28 U.S.C. §§ 1331 and 1338(a).

4. This Court has personal jurisdiction over Defendant in this action because Defendant has committed acts within the Western District of Texas giving rise to this action and has established minimum contacts with this forum such that the exercise of jurisdiction over Defendant would not offend traditional notions of fair play and substantial justice. Defendant, directly and through subsidiaries or intermediaries, has committed and continues to commit acts of infringement in this District by, among other things, offering to sell and selling products and/or services that infringe the asserted patents.

5. Venue is proper in this district under 28 U.S.C. § 1400(b). Defendant has established places of business in the Western District of Texas. Defendant is registered to do business in Texas. Upon information and belief, Defendant has transacted business in this District and has committed acts of direct and indirect infringement in this District.

COUNT I

INFRINGEMENT OF U.S. PATENT NO. 10,277,675

6. Data Scape is the owner by assignment of United States Patent No. 10,277,675 ("the '675 Patent"), entitled "Communication System And Its Method and Communication Apparatus And Its Method." The '675 Patent was duly and legally issued by the United States Patent and Trademark Office on April 30, 2019. A true and correct copy of the '675 Patent is included as Exhibit A.

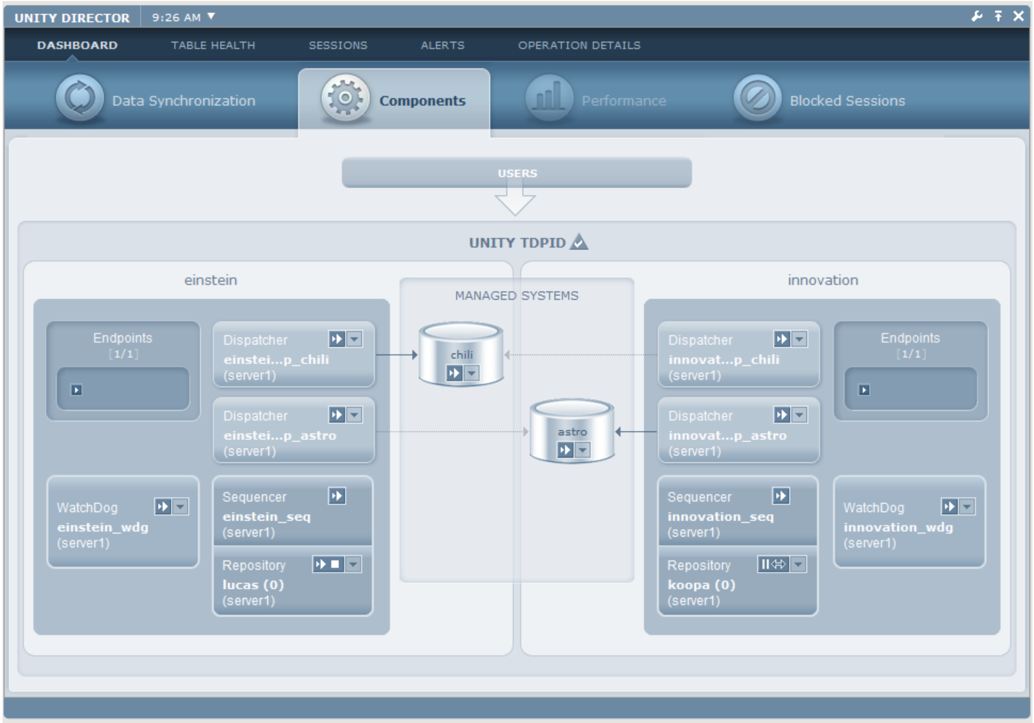
7. Defendant has offered for sale, sold and/or imported into the United States products and services that infringe the ’675 patent, and continues to do so. By way of illustrative example, these infringing products and services include, without limitation, Defendant’s products and services, *e.g.*, Unity, Unity Director, Unity Loader, Data Mover, Teradata Managed Server Unity, Unity Director/Loader Model 8-81X and Expansion Server Model 8-81XE, Unity Source Link Server, Teradata Managed Server Unity Data Mover, and all versions and variations thereof since the issuance of the ’675 Patent (“Accused Instrumentalities”).

8. Defendant has directly infringed and continues to infringe the ’675 Patent, for example, by making, selling, offering for sale, and/or importing the Accused Instrumentalities, and through its own use and testing of the Accused Instrumentalities. Defendant uses the Accused Instrumentalities for its own internal non-testing business purposes, while testing the Accused Instrumentalities, and while providing technical support and repair services for the Accused Instrumentalities to its customers.

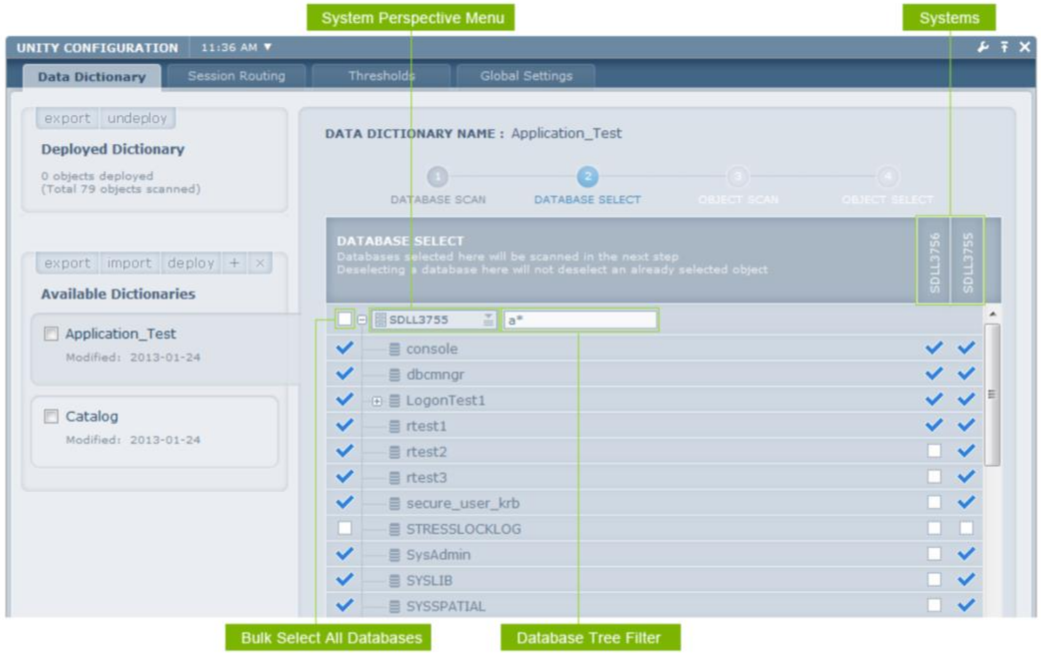
9. For example, the Accused Instrumentalities infringe Claim 1 (as well as other claims) of the ’675 Patent. One non-limiting example of the Accused Instrumentalities’ infringement is presented below:

10. The Accused Instrumentalities include “a communication system including a first apparatus having a first hardware storage medium, and a second apparatus.” For example, the Accused Instrumentalities communicate data stored on one device (e.g. a device running Unity, Unity Director, Unity Loader, Teradata Bulk Transfer, Teradata Data Mover) to another device with a storage medium (e.g. a Teradata Database system). See, e.g., Teradata Unity Datasheet (EB7192.pdf), available at

http://assets.teradata.com/resourceCenter/downloads/Datasheets/EB7192.pdf. ("Another benefit of Teradata Unity is the capability to synchronize multiple Teradata systems. The SQL Multicast feature delivers SQL commands to all participating systems in the Teradata Analytical Ecosystem. Teradata Unity will automatically queue up and dispatch the incoming SQL commands in the order in which they're received, maintaining consistency and integrity across systems."); http://downloads.teradata.com/uda/articles/unity-director-initial-configuration-and-setup:



b. Select databases used in the application:



Deployed Dictionary

0 objects deployed
(Total 79 objects scanned)

Available Dictionaries

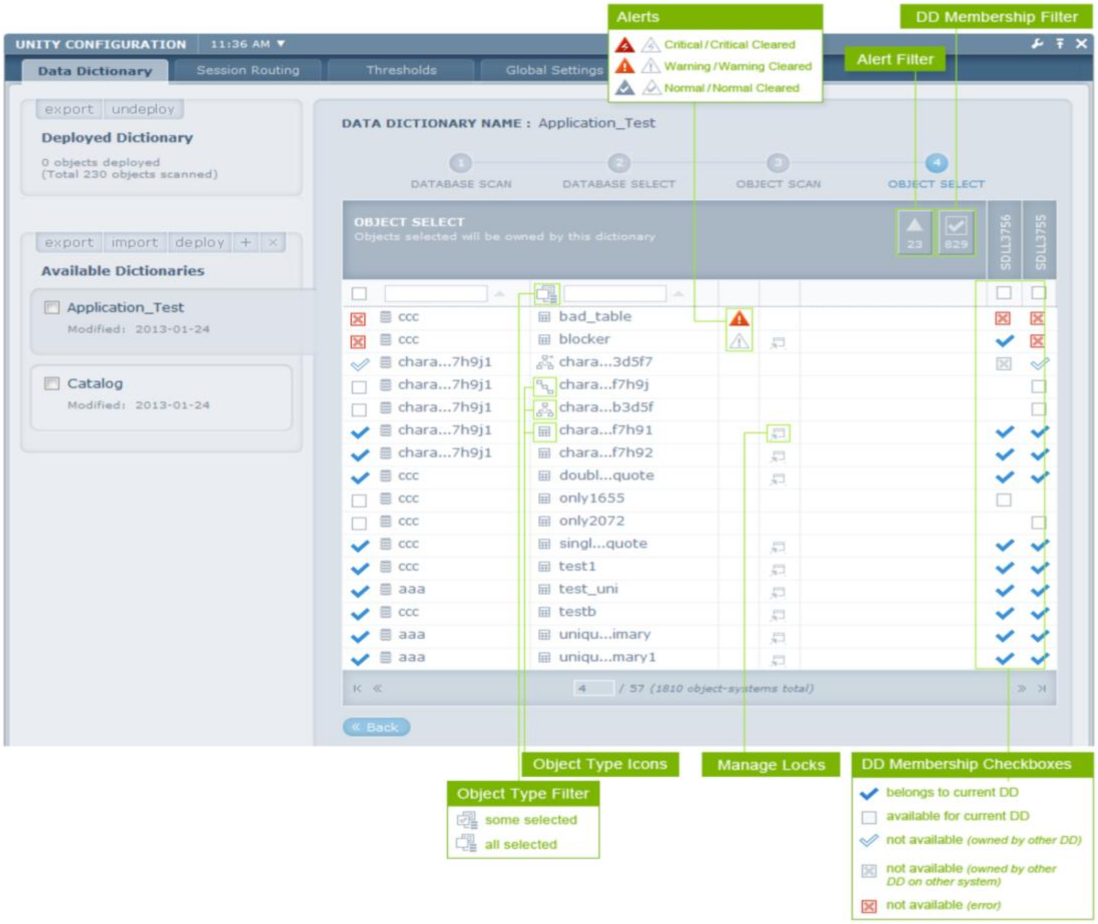
□ Application_Test
Modified: 2013-01-24

□ Catalog
Modified: 2013-01-24

DATA DICTIONARY NAME : Application_Test

DATABASE SELECT
Databases selected here will be scanned in the next step
Deselecting a database here will not deselect an already selected object

□	SDLL3755	a*	SDLL3756	SDLL3755
✓	console		✓	✓
✓	dbcmngr		✓	✓
✓	LogonTest1		✓	✓
✓	rtest1		✓	✓
✓	rtest2		□	✓
✓	rtest3		□	✓
✓	secure_user_krb		□	✓
□	STRESSLOCKLOG		□	□
✓	SysAdmin		□	✓
✓	SYSLIB		□	✓
✓	SYSSPATIAL		□	✓



Deployed Dictionary

0 objects deployed
(Total 230 objects scanned)

Available Dictionaries

□ Application_Test
Modified: 2013-01-24

□ Catalog
Modified: 2013-01-24

DATA DICTIONARY NAME : Application_Test

OBJECT SELECT
Objects selected will be owned by this dictionary

□					SDLL3756	SDLL3755
☒	ccc	bad_table	▲		☒	☒
☒	ccc	blocker	△	🔒	✓	☒
✓	chara...7h9j1	chara...3d5f7			☒	✓
□	chara...7h9j1	chara...f7h9j				□
□	chara...7h9j1	chara...b3d5f				□
✓	chara...7h9j1	chara...f7h91		🔒	✓	✓
✓	chara...7h9j1	chara...f7h92		🔒	✓	✓
✓	ccc	doubl...quote		🔒	✓	✓
□	ccc	only1655			□	
□	ccc	only2072				□
✓	ccc	singl...quote		🔒	✓	✓
✓	ccc	test1		🔒	✓	✓
✓	aaa	test_uni		🔒	✓	✓
✓	ccc	testb		🔒	✓	✓
✓	aaa	uniqu...imary		🔒	✓	✓
✓	aaa	uniqu...mary1		🔒	✓	✓

4 / 57 (1810 object-systems total)

11. The Accused Instrumentalities include “a second apparatus comprising a second hardware storage medium configured to store management information of data to be transferred to said first storage medium.” For example, the Accused Instrumentalities include disk drives and/or solid state disks, which are necessary for the operation of the Accused Instrumentalities. *See, e.g.*, Teradata Managed Server for Cabinet and Server Models 4X5, 4X7, and 6X7 Product and Site Preparation Guide, *available at* https://docs.teradata.com/reader/DdAmXXaBagDQwtH6uaApUg/ce~Xolu7GagxHN5mOr4eRQ (“The following table outlines the features of the Teradata Managed Server 687 Unity models. *** Disk drives: Six 3.5-inch 450 GB or 600 GB SAS 15K RPM hard drives”); Teradata Unity User Guide, Release 15.10 (2520-066K.pdf) (“Unity supports different sized Teradata systems that load at different rates and minimizes impact on the Bulk Load client when a system is offline. Unity manages an internal disk array to accommodate the disk space requirements for landing data before loading the data to Teradata systems. Data is landed to disk for any of the following reasons: • Data cannot be loaded to one or more of the target systems due to target, error, or restart tables in the Interrupted state or an unavailable target system. • A data checkpoint did not occur on all systems, and there is insufficient memory to store the data in memory. • A data checkpoint is successful on one system, but not on the others. Unity deletes landed data stored on the disk array after the data successfully loads to the Teradata systems, and the begin loading record of the Bulk Load job is overwritten in the Recovery Log. Management of the disk space by Unity reduces the total amount of disk space required for the load files.”).

12. The Accused Instrumentalities include “a second apparatus comprising a hardware interface configured to communicate data with said first apparatus.” For example,

the Accused Instrumentalities communicate with external Teradata database servers. See, e.g., EB7192.pdf ("Another benefit of Teradata Unity is the capability to synchronize multiple Teradata systems. The SQL Multicast feature delivers SQL commands to all participating systems in the Teradata Analytical Ecosystem. Teradata Unity will automatically queue up and dispatch the incoming SQL commands in the order in which they're received, maintaining consistency and integrity across systems."); *id.*:

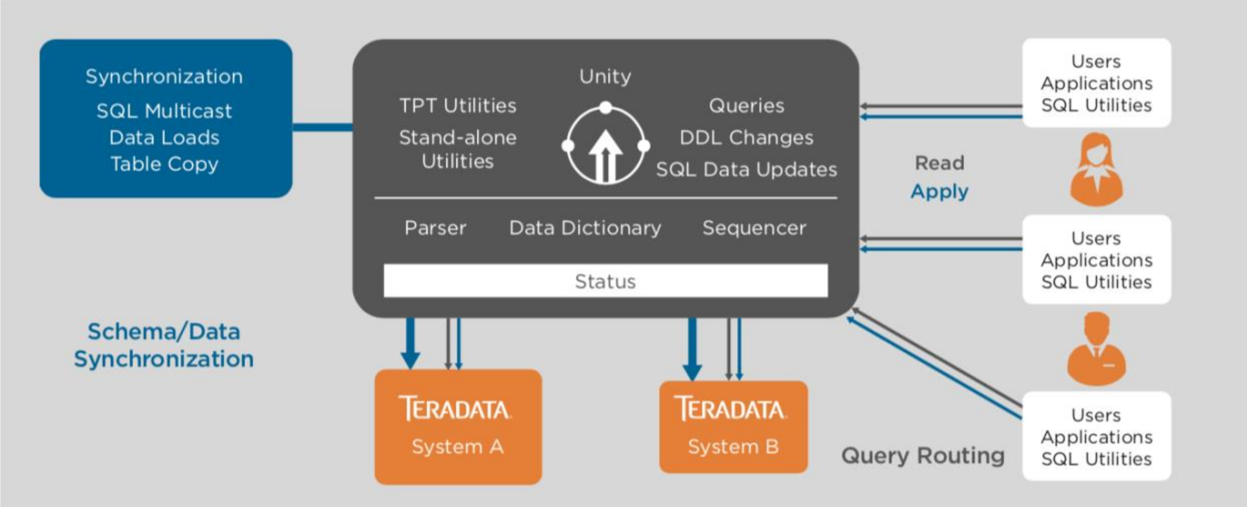
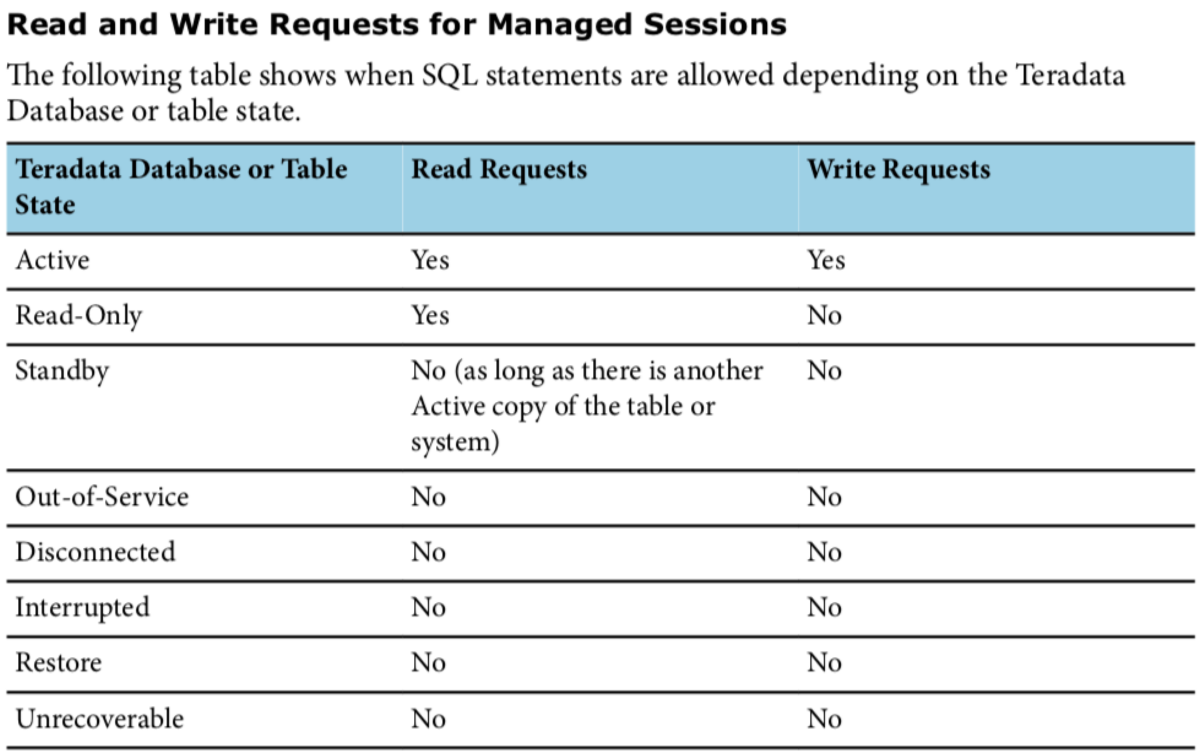


Figure 1. Unity—Query Routing and Data Synchronization.

13. The Accused Instrumentalities include "a second apparatus comprising a processor configured to detect whether said first apparatus and said second apparatus are connected." For example, the Accused Instrumentalities detects the connection state with the Teradata Database and transfers data only when the database is in "Active" or "Read-Only" state, i.e. connected, and not when the Teradata Database is in "Disconnected" state. See, e.g., 2520-066K.pdf:



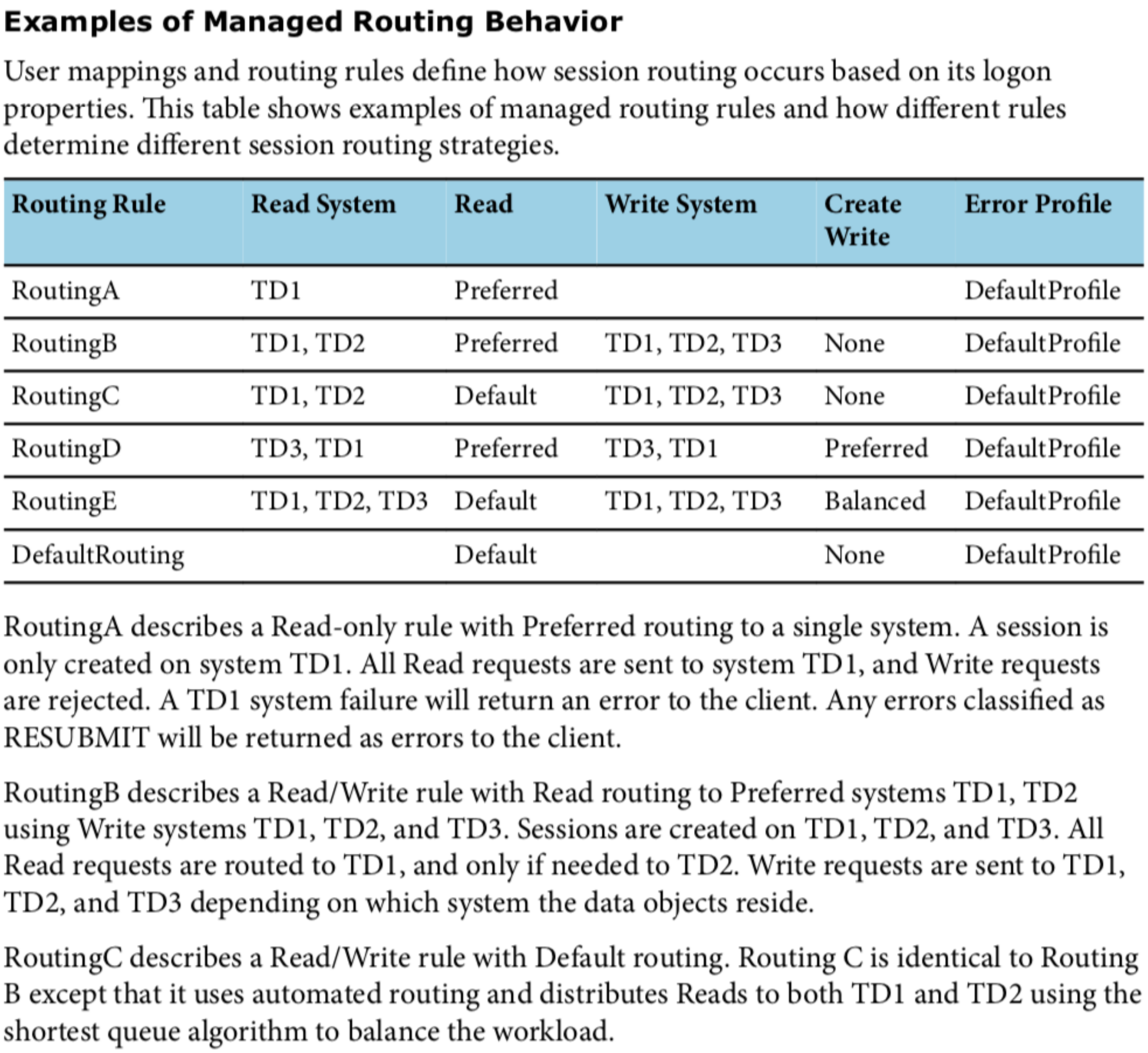
Read and Write Requests for Managed Sessions

The following table shows when SQL statements are allowed depending on the Teradata Database or table state.

Teradata Database or Table State	Read Requests	Write Requests
Active	Yes	Yes
Read-Only	Yes	No
Standby	No (as long as there is another Active copy of the table or system)	No
Out-of-Service	No	No
Disconnected	No	No
Interrupted	No	No
Restore	No	No
Unrecoverable	No	No

14. The Accused Instrumentalities include "a second apparatus comprising a processor configured to select certain data to be transferred." For example, the Accused Instrumentalities include an editor to select data to send to individual Teradata database

systems based on, *e.g.*, managed routing rules, and edits management information based on the selection. *See*, *e.g.*, 2520-066K.pdf:



Examples of Managed Routing Behavior

User mappings and routing rules define how session routing occurs based on its logon properties. This table shows examples of managed routing rules and how different rules determine different session routing strategies.

Routing Rule	Read System	Read	Write System	Create Write	Error Profile
RoutingA	TD1	Preferred			DefaultProfile
RoutingB	TD1, TD2	Preferred	TD1, TD2, TD3	None	DefaultProfile
RoutingC	TD1, TD2	Default	TD1, TD2, TD3	None	DefaultProfile
RoutingD	TD3, TD1	Preferred	TD3, TD1	Preferred	DefaultProfile
RoutingE	TD1, TD2, TD3	Default	TD1, TD2, TD3	Balanced	DefaultProfile
DefaultRouting		Default		None	DefaultProfile

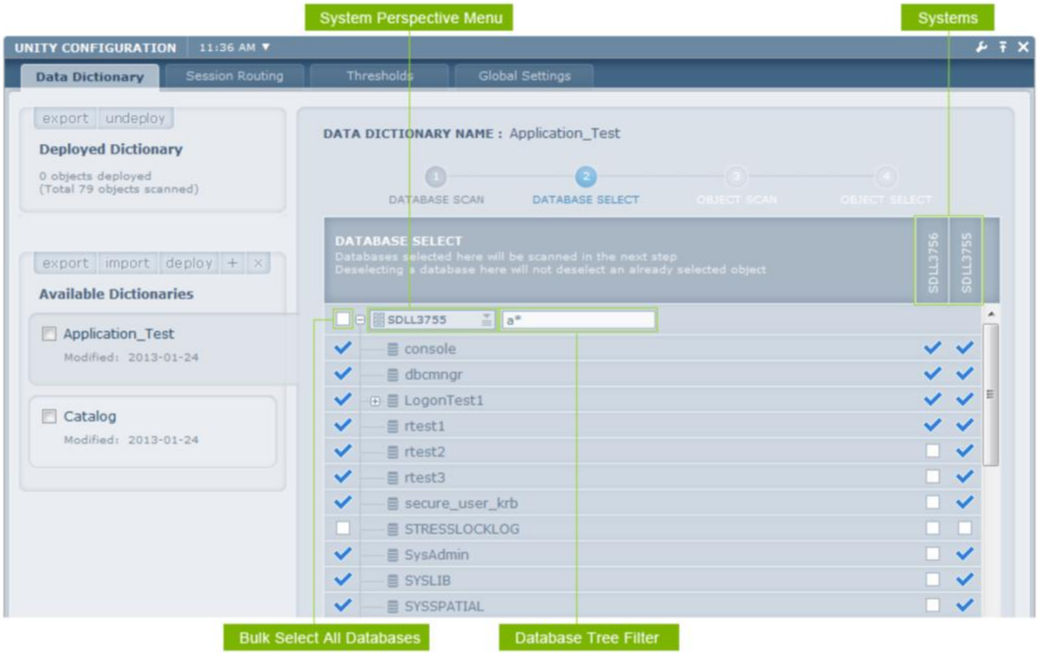
RoutingA describes a Read-only rule with Preferred routing to a single system. A session is only created on system TD1. All Read requests are sent to system TD1, and Write requests are rejected. A TD1 system failure will return an error to the client. Any errors classified as RESUBMIT will be returned as errors to the client.

RoutingB describes a Read/Write rule with Read routing to Preferred systems TD1, TD2 using Write systems TD1, TD2, and TD3. Sessions are created on TD1, TD2, and TD3. All Read requests are routed to TD1, and only if needed to TD2. Write requests are sent to TD1, TD2, and TD3 depending on which system the data objects reside.

RoutingC describes a Read/Write rule with Default routing. Routing C is identical to Routing B except that it uses automated routing and distributes Reads to both TD1 and TD2 using the shortest queue algorithm to balance the workload.

See also http://downloads.teradata.com/uda/articles/unity-director-initial-configuration-and-setup:

b. Select databases used in the application:



Deployed Dictionary

0 objects deployed
(Total 79 objects scanned)

Available Dictionaries

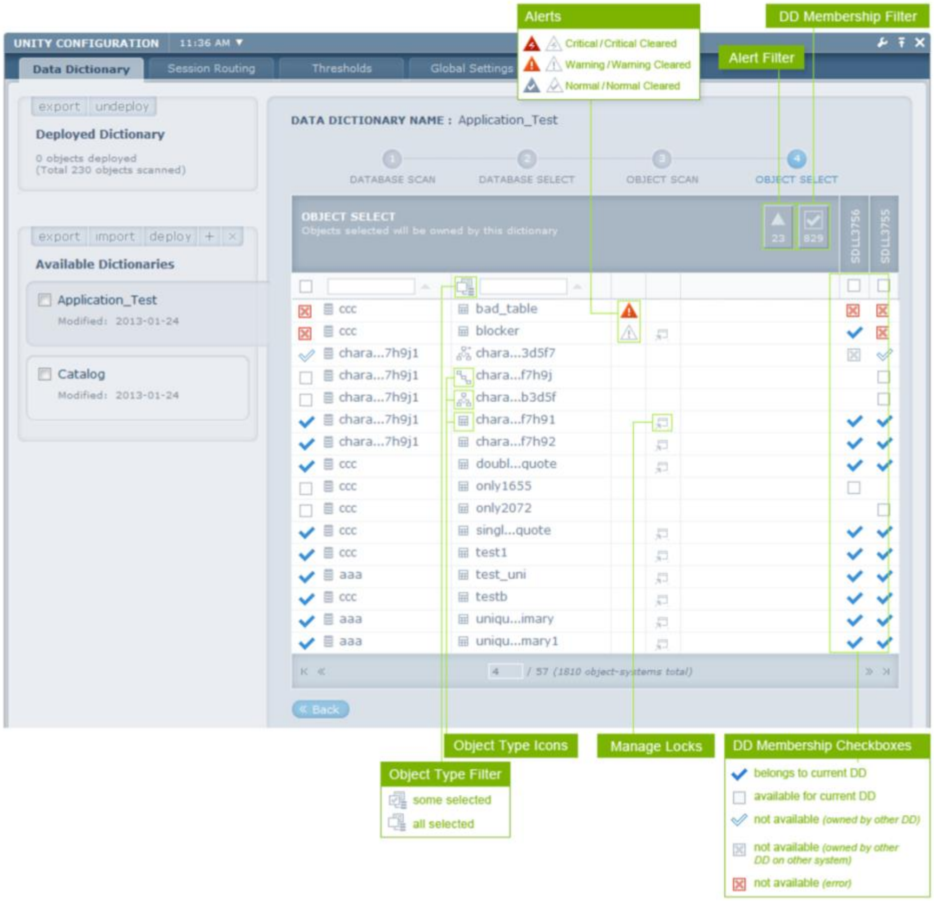
□ Application_Test
Modified: 2013-01-24

□ Catalog
Modified: 2013-01-24

DATA DICTIONARY NAME : Application_Test

DATABASE SELECT
Databases selected here will be scanned in the next step
Deselecting a database here will not deselect an already selected object

□	SDLL3755	a*	SDLL3756	SDLL3755
✓	console		✓	✓
✓	dbcmngr		✓	✓
✓	LogonTest1		✓	✓
✓	rtest1		✓	✓
✓	rtest2		□	✓
✓	rtest3		□	✓
✓	secure_user_krb		□	✓
□	STRESSLOCKLOG		□	□
✓	SysAdmin		□	✓
✓	SYSLIB		□	✓
✓	SYSSPATIAL		□	✓



Deployed Dictionary

0 objects deployed
(Total 230 objects scanned)

Available Dictionaries

□ Application_Test
Modified: 2013-01-24

□ Catalog
Modified: 2013-01-24

DATA DICTIONARY NAME : Application_Test

OBJECT SELECT
Objects selected will be owned by this dictionary

▲ 23 ☑ 829

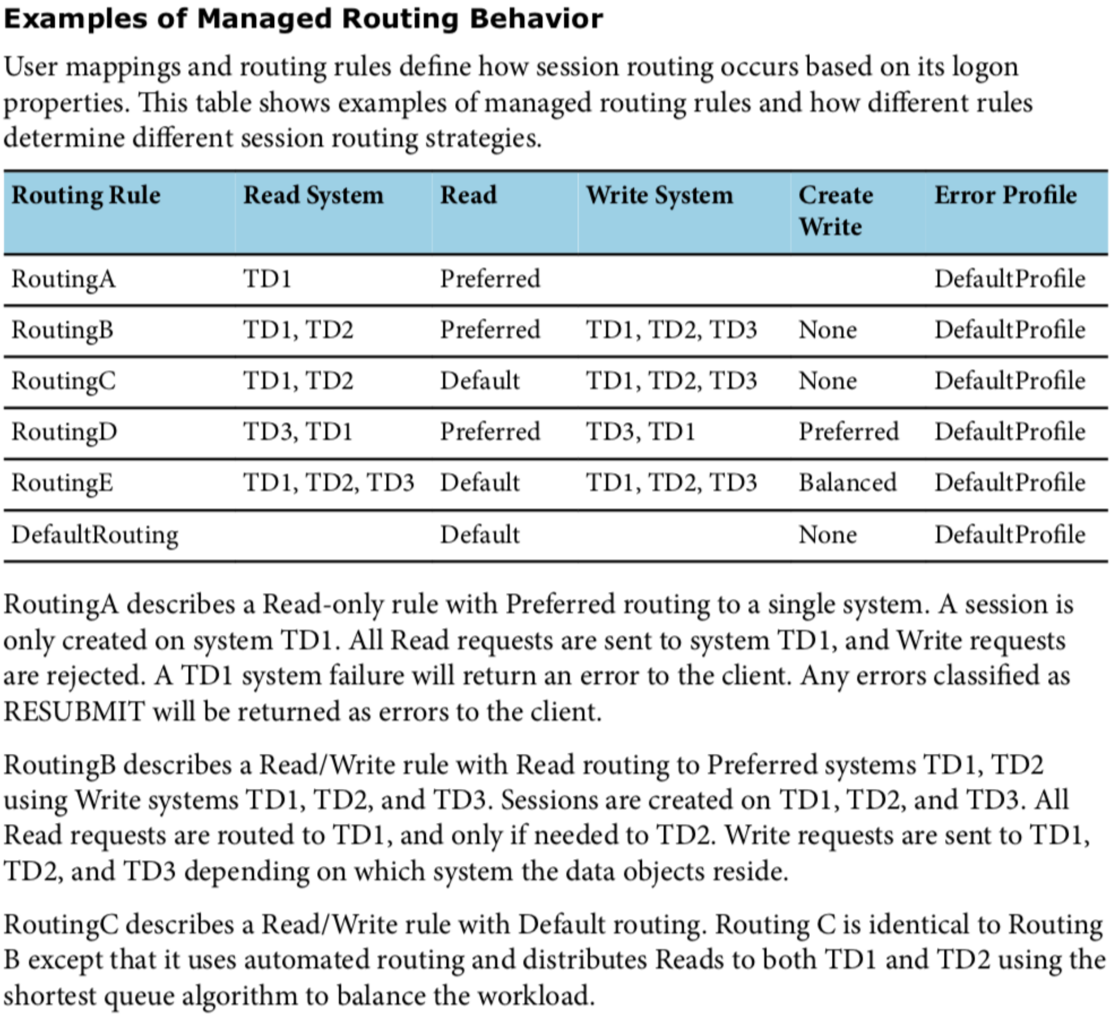
□					SDLL3756	SDLL3755
☒	ccc	bad_table	▲		☒	☒
☒	ccc	blocker	△	🔒	✓	☒
✓	chara...7h9j1	chara...3d5f7			☒	✓
□	chara...7h9j1	chara...f7h9j				□
□	chara...7h9j1	chara...b3d5f				□
✓	chara...7h9j1	chara...f7h91		🔒	✓	✓
✓	chara...7h9j1	chara...f7h92		🔒	✓	✓
✓	ccc	doubl...quote		🔒	✓	✓
□	ccc	only1655			□	
□	ccc	only2072				□
✓	ccc	singl...quote		🔒	✓	✓
✓	ccc	test1		🔒	✓	✓
✓	aaa	test_uni		🔒	✓	✓
✓	ccc	testb		🔒	✓	✓
✓	aaa	uniqu...imary		🔒	✓	✓
✓	aaa	uniqu...mary1		🔒	✓	✓

4 / 57 *(1810 object-systems total)*

Object Type Filter
- some selected
- all selected

DD Membership Checkboxes
- ✓ belongs to current DD
- □ available for current DD
- ✓ not available *(owned by other DD)*
- ☒ not available *(owned by other DD on other system)*
- ☒ not available *(error)*

15. The Accused Instrumentalities include "a second apparatus comprising a processor configured to edit said management information based on said selection without regard to the connection of said first apparatus and said second apparatus." For example, the Accused Instrumentalities include an editor to select data to send to individual Teradata database systems based on, *e.g.*, managed routing rules, and edits management information based on the selection. This editing is done without regard to the connection between the Accused Instrumentalities and the target database system, *e.g.*, when the Accused Instrumentalities receive commands to create or drop tables using SQL Data Definition Language. *See, e.g.*, 2520-066K.pdf:



Examples of Managed Routing Behavior

User mappings and routing rules define how session routing occurs based on its logon properties. This table shows examples of managed routing rules and how different rules determine different session routing strategies.

Routing Rule	Read System	Read	Write System	Create Write	Error Profile
RoutingA	TD1	Preferred			DefaultProfile
RoutingB	TD1, TD2	Preferred	TD1, TD2, TD3	None	DefaultProfile
RoutingC	TD1, TD2	Default	TD1, TD2, TD3	None	DefaultProfile
RoutingD	TD3, TD1	Preferred	TD3, TD1	Preferred	DefaultProfile
RoutingE	TD1, TD2, TD3	Default	TD1, TD2, TD3	Balanced	DefaultProfile
DefaultRouting		Default		None	DefaultProfile

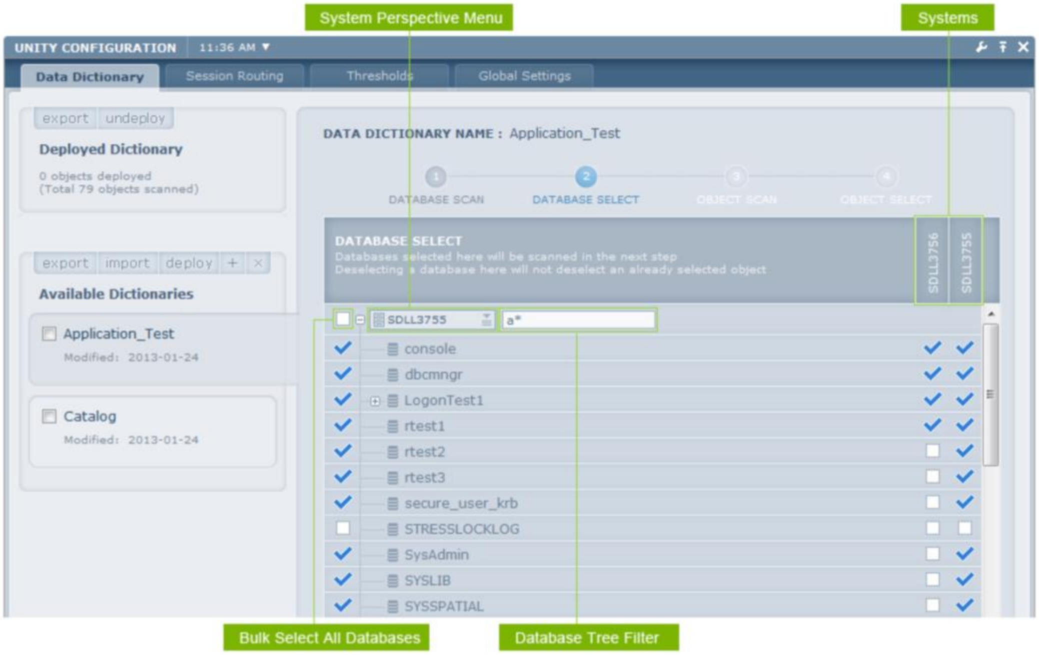
RoutingA describes a Read-only rule with Preferred routing to a single system. A session is only created on system TD1. All Read requests are sent to system TD1, and Write requests are rejected. A TD1 system failure will return an error to the client. Any errors classified as RESUBMIT will be returned as errors to the client.

RoutingB describes a Read/Write rule with Read routing to Preferred systems TD1, TD2 using Write systems TD1, TD2, and TD3. Sessions are created on TD1, TD2, and TD3. All Read requests are routed to TD1, and only if needed to TD2. Write requests are sent to TD1, TD2, and TD3 depending on which system the data objects reside.

RoutingC describes a Read/Write rule with Default routing. Routing C is identical to Routing B except that it uses automated routing and distributes Reads to both TD1 and TD2 using the shortest queue algorithm to balance the workload.

See also http://downloads.teradata.com/uda/articles/unity-director-initial-configuration-and-setup:

b. Select databases used in the application:



0 objects deployed
(Total 79 objects scanned)

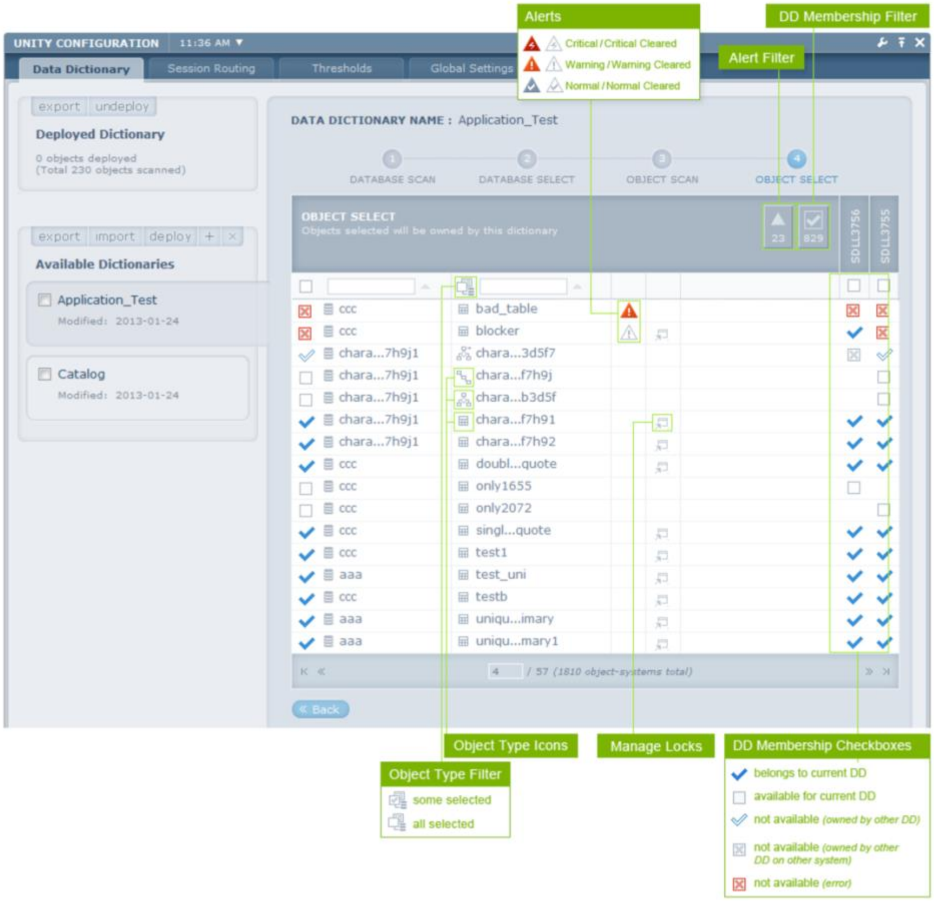
☐ Application_Test
Modified: 2013-01-24

☐ Catalog
Modified: 2013-01-24

DATA DICTIONARY NAME : Application_Test

DATABASE SELECT
Databases selected here will be scanned in the next step
Deselecting a database here will not deselect an already selected object

☐	SDLL3755 a*	SDLL3756	SDLL3755
✓	console	✓	✓
✓	dbcmngr	✓	✓
✓	LogonTest1	✓	✓
✓	rtest1	✓	✓
✓	rtest2	☐	✓
✓	rtest3	☐	✓
✓	secure_user_krb	☐	✓
☐	STRESSLOCKLOG	☐	☐
✓	SysAdmin	☐	✓
✓	SYSLIB	☐	✓
✓	SYSSPATIAL	☐	✓



Alerts
- ⚠ △ Critical / Critical Cleared
- ⚠ △ Warning / Warning Cleared
- ▲ △ Normal / Normal Cleared

0 objects deployed
(Total 230 objects scanned)

☐ Application_Test
Modified: 2013-01-24

☐ Catalog
Modified: 2013-01-24

DATA DICTIONARY NAME : Application_Test

OBJECT SELECT
Objects selected will be owned by this dictionary

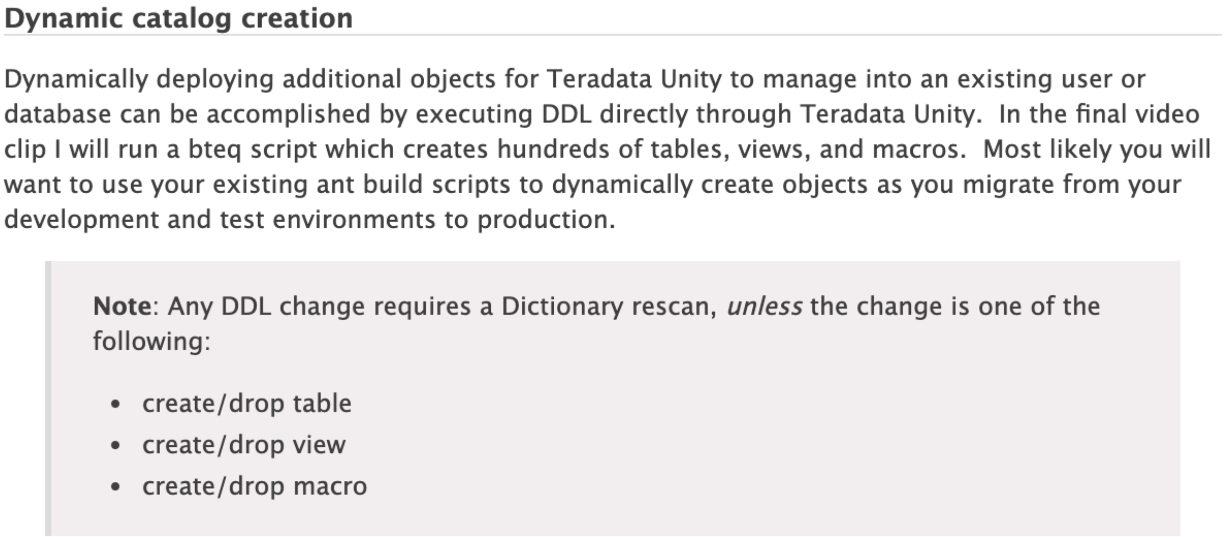
☐					SDLL3756 ☐	SDLL3755 ☐
☒	ccc	bad_table	⚠		☒	☒
☒	ccc	blocker	△	🔒	✓	☒
✓	chara...7h9j1	chara...3d5f7			☒	✓
☐	chara...7h9j1	chara...f7h9j				☐
☐	chara...7h9j1	chara...b3d5f				☐
✓	chara...7h9j1	chara...f7h91		🔒	✓	✓
✓	chara...7h9j1	chara...f7h92		🔒	✓	✓
✓	ccc	doubl...quote		🔒	✓	✓
☐	ccc	only1655			☐	
☐	ccc	only2072				☐
✓	ccc	singl...quote		🔒	✓	✓
✓	ccc	test1		🔒	✓	✓
✓	aaa	test_uni		🔒	✓	✓
✓	ccc	testb		🔒	✓	✓
✓	aaa	uniqu...imary		🔒	✓	✓
✓	aaa	uniqu...mary1		🔒	✓	✓

K « 4 / 57 (1810 object-systems total) » >I

Object Type Filter
- some selected
- all selected

DD Membership Checkboxes
- ✓ belongs to current DD
- ☐ available for current DD
- ✓ not available *(owned by other DD)*
- ☒ not available *(owned by other DD on other system)*
- ☒ not available *(error)*

See also http://developer.teradata.com/uda/articles/unity-13-10-foundation-catalog-deploy-examples:

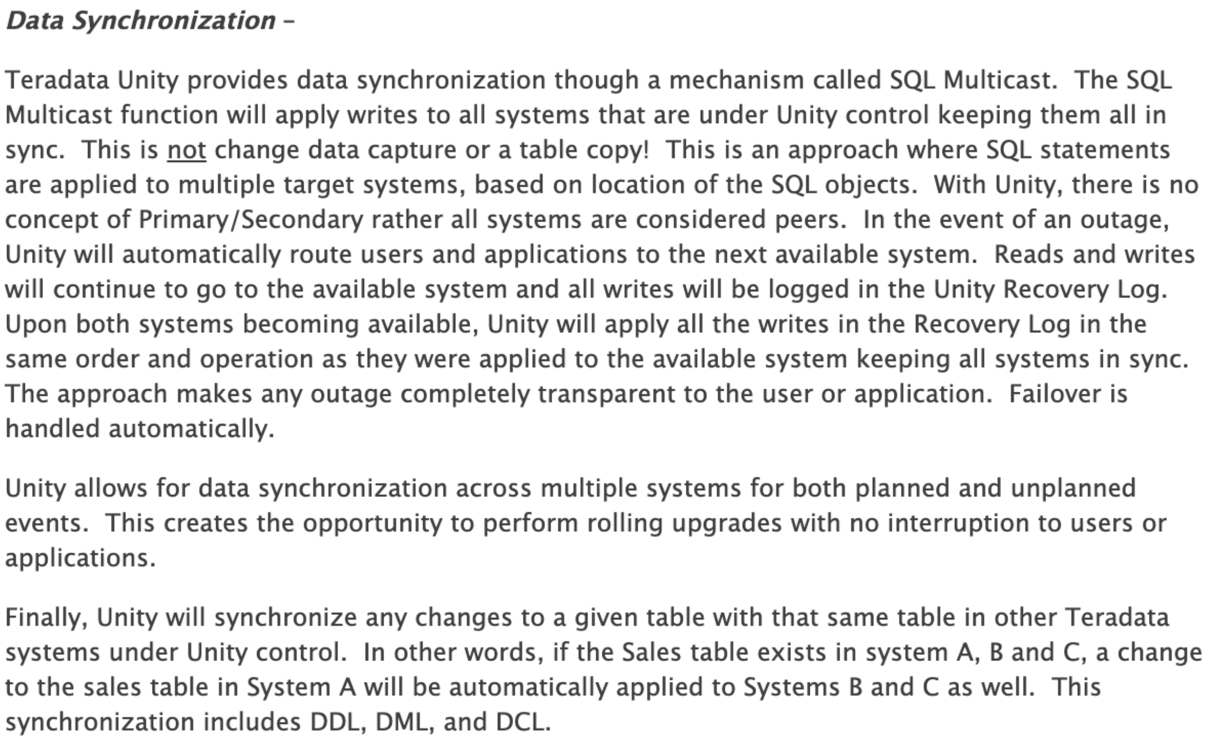


Dynamic catalog creation

Dynamically deploying additional objects for Teradata Unity to manage into an existing user or database can be accomplished by executing DDL directly through Teradata Unity. In the final video clip I will run a bteq script which creates hundreds of tables, views, and macros. Most likely you will want to use your existing ant build scripts to dynamically create objects as you migrate from your development and test environments to production.

> **Note**: Any DDL change requires a Dictionary rescan, *unless* the change is one of the following:
>
> - create/drop table
> - create/drop view
> - create/drop macro

See also https://developer.teradata.com/uda/articles/stay-in-sync-introducing-teradata-unity-13-10:



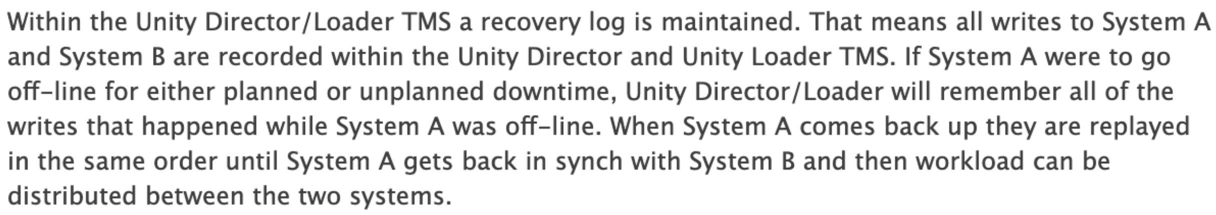
Data Synchronization –

Teradata Unity provides data synchronization though a mechanism called SQL Multicast. The SQL Multicast function will apply writes to all systems that are under Unity control keeping them all in sync. This is not change data capture or a table copy! This is an approach where SQL statements are applied to multiple target systems, based on location of the SQL objects. With Unity, there is no concept of Primary/Secondary rather all systems are considered peers. In the event of an outage, Unity will automatically route users and applications to the next available system. Reads and writes will continue to go to the available system and all writes will be logged in the Unity Recovery Log. Upon both systems becoming available, Unity will apply all the writes in the Recovery Log in the same order and operation as they were applied to the available system keeping all systems in sync. The approach makes any outage completely transparent to the user or application. Failover is handled automatically.

Unity allows for data synchronization across multiple systems for both planned and unplanned events. This creates the opportunity to perform rolling upgrades with no interruption to users or applications.

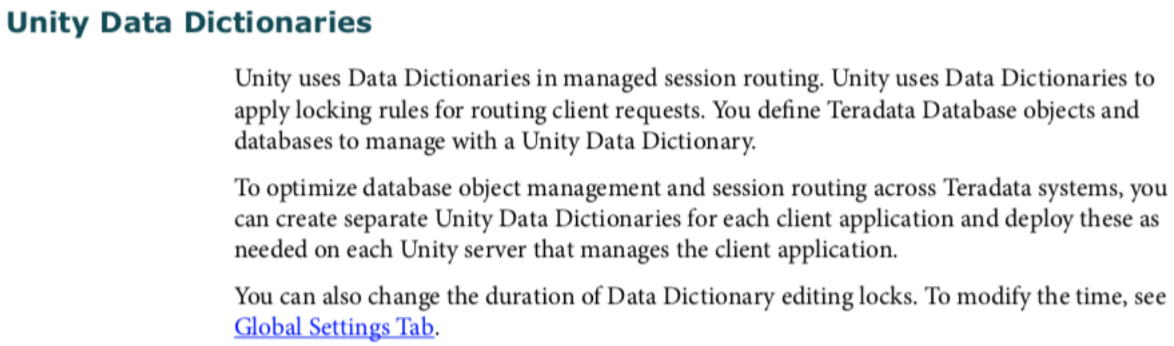
Finally, Unity will synchronize any changes to a given table with that same table in other Teradata systems under Unity control. In other words, if the Sales table exists in system A, B and C, a change to the sales table in System A will be automatically applied to Systems B and C as well. This synchronization includes DDL, DML, and DCL.

See also, e.g., https://developer.teradata.com/general/articles/multi-active-systems-with-new-unity-directorloader-14-11:



Within the Unity Director/Loader TMS a recovery log is maintained. That means all writes to System A and System B are recorded within the Unity Director and Unity Loader TMS. If System A were to go off-line for either planned or unplanned downtime, Unity Director/Loader will remember all of the writes that happened while System A was off-line. When System A comes back up they are replayed in the same order until System A gets back in synch with System B and then workload can be distributed between the two systems.

16. The Accused Instrumentalities include "a second apparatus comprising a processor configured to compare said management information edited by said processor with management information of data stored in said first storage medium." For example, the Accused Instrumentalities provide a mechanism to manage 'Data Dictionaries' that define what data/tables should be synced to a given database server. See, e.g., Teradata Unity User Guide, Release 15.10 (2520-066K.pdf):



Unity Data Dictionaries

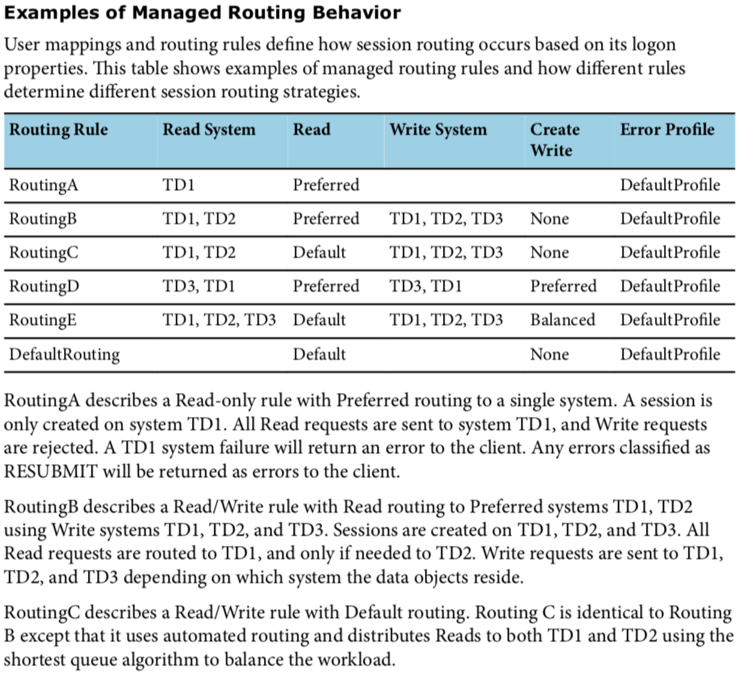
Unity uses Data Dictionaries in managed session routing. Unity uses Data Dictionaries to apply locking rules for routing client requests. You define Teradata Database objects and databases to manage with a Unity Data Dictionary.

To optimize database object management and session routing across Teradata systems, you can create separate Unity Data Dictionaries for each client application and deploy these as needed on each Unity server that manages the client application.

You can also change the duration of Data Dictionary editing locks. To modify the time, see Global Settings Tab.

17. The Accused Instrumentalities include "a second apparatus comprising a processor configured to transmit the selected data stored in said second apparatus to said first apparatus via said hardware interface based on said management information edited by said processor when said processor detects that said first apparatus and said second apparatus are connected based upon a result of the comparison." For example, the Accused Instrumentalities transmits data to the Teradata Database server to an external database system that is identified, when it is connected by a unique identifier called a TDPID, and

the Accused Instrumentalities controls what data to transmit based on, e.g., managed routing rules, and edits management information based on the selection. See, e.g., 2520-066K.pdf ("DICTIONARY SET PREFERRED: Purpose: Sets the Preferred Write system to the Teradata Database system identified by the TDPID value. To set the Preferred Write system for all tables in a database, specify only the name of the database. To set the Preferred Write system for a single table, specify both the database name and the table name. *** tdpid: Unique identifier (TDPID) of a Teradata Database system."); EB7192.pdf ("Taken together, the data loading and query management work as one to sequence database changes and query requests. This ensures that data updates and database structure changes are always applied in the order received; automatically coordinating and maintaining order and consistency across systems. And, it is specifically designed to work with systems that are not identical. For example, if the primary integrated data warehouse contains 100 percent of the data, and there is a second system for high availability that holds a 30 percent subset of that data, Teradata Unity understands the capabilities and limitations of each system and will route data loads accordingly."); *id.*:



Examples of Managed Routing Behavior

User mappings and routing rules define how session routing occurs based on its logon properties. This table shows examples of managed routing rules and how different rules determine different session routing strategies.

Routing Rule	Read System	Read	Write System	Create Write	Error Profile
RoutingA	TD1	Preferred			DefaultProfile
RoutingB	TD1, TD2	Preferred	TD1, TD2, TD3	None	DefaultProfile
RoutingC	TD1, TD2	Default	TD1, TD2, TD3	None	DefaultProfile
RoutingD	TD3, TD1	Preferred	TD3, TD1	Preferred	DefaultProfile
RoutingE	TD1, TD2, TD3	Default	TD1, TD2, TD3	Balanced	DefaultProfile
DefaultRouting		Default		None	DefaultProfile

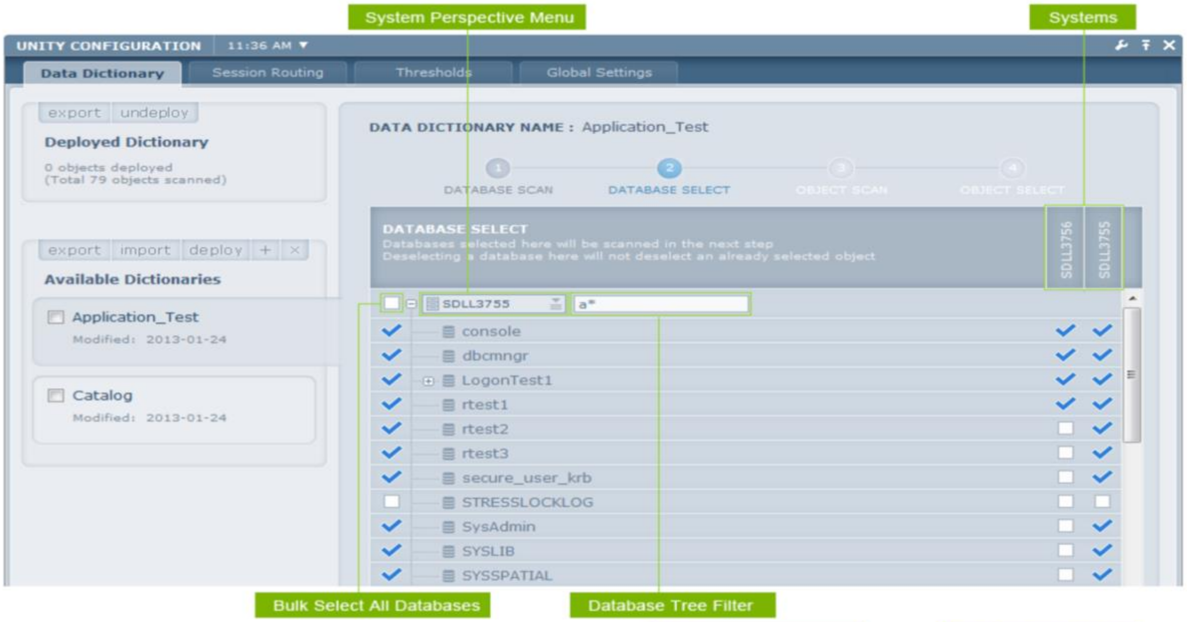
RoutingA describes a Read-only rule with Preferred routing to a single system. A session is only created on system TD1. All Read requests are sent to system TD1, and Write requests are rejected. A TD1 system failure will return an error to the client. Any errors classified as RESUBMIT will be returned as errors to the client.

RoutingB describes a Read/Write rule with Read routing to Preferred systems TD1, TD2 using Write systems TD1, TD2, and TD3. Sessions are created on TD1, TD2, and TD3. All Read requests are routed to TD1, and only if needed to TD2. Write requests are sent to TD1, TD2, and TD3 depending on which system the data objects reside.

RoutingC describes a Read/Write rule with Default routing. Routing C is identical to Routing B except that it uses automated routing and distributes Reads to both TD1 and TD2 using the shortest queue algorithm to balance the workload.

See also http://downloads.teradata.com/uda/articles/unity-director-initial-configuration-and-setup:

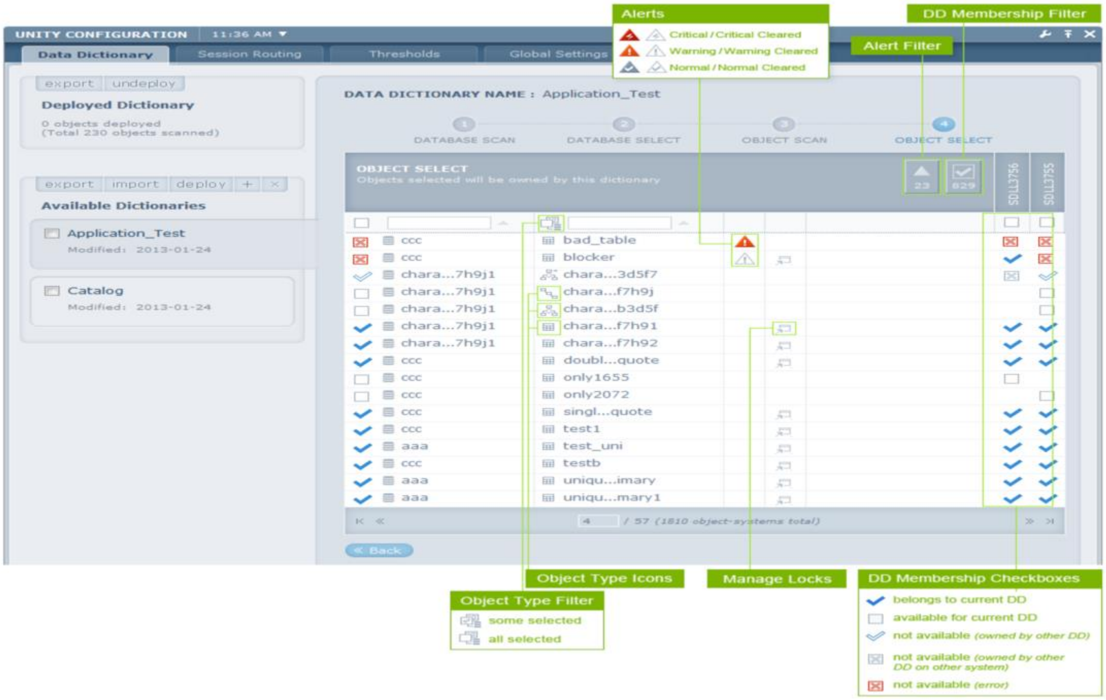
b. Select databases used in the application:



DATA DICTIONARY NAME : Application_Test

Databases selected here will be scanned in the next step
Deselecting a database here will not deselect an already selected object

□	SDLL3755	SDLL3756	SDLL3755
	a*		
✓	console	✓	✓
✓	dbcmngr	✓	✓
✓	LogonTest1	✓	✓
✓	rtest1	✓	✓
✓	rtest2	□	✓
✓	rtest3	□	✓
✓	secure_user_krb	□	✓
□	STRESSLOCKLOG	□	□
✓	SysAdmin	□	✓
✓	SYSLIB	□	✓
✓	SYSSPATIAL	□	✓



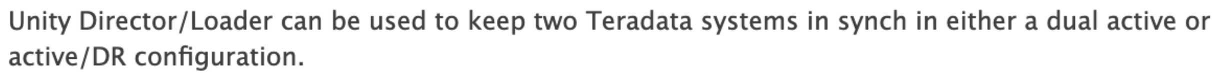
DATA DICTIONARY NAME : Application_Test

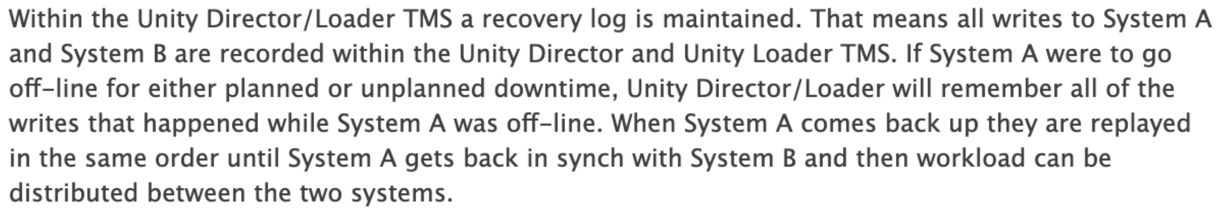
Objects selected will be owned by this dictionary

□					SDLL3756	SDLL3755
					□	□
☒	ccc	bad_table	▲		☒	☒
☒	ccc	blocker	△	🔒	✓	☒
✓	chara...7h9j1	chara...3d5f7			☒	✓
□	chara...7h9j1	chara...f7h9j				□
□	chara...7h9j1	chara...b3d5f				□
✓	chara...7h9j1	chara...f7h91		🔒	✓	✓
✓	chara...7h9j1	chara...f7h92		🔒	✓	✓
✓	ccc	doubl...quote		🔒	✓	✓
□	ccc	only1655			□	
□	ccc	only2072				□
✓	ccc	singl...quote		🔒	✓	✓
✓	ccc	test1		🔒	✓	✓
✓	aaa	test_uni		🔒	✓	✓
✓	ccc	testb		🔒	✓	✓
✓	aaa	uniqu...imary		🔒	✓	✓
✓	aaa	uniqu...mary1		🔒	✓	✓

4 / 57 (1810 object-systems total)

See also, e.g., https://developer.teradata.com/general/articles/multi-active-systems-with-new-unity-directorloader-14-11:


Unity Director/Loader can be used to keep two Teradata systems in synch in either a dual active or active/DR configuration.


Within the Unity Director/Loader TMS a recovery log is maintained. That means all writes to System A and System B are recorded within the Unity Director and Unity Loader TMS. If System A were to go off-line for either planned or unplanned downtime, Unity Director/Loader will remember all of the writes that happened while System A was off-line. When System A comes back up they are replayed in the same order until System A gets back in synch with System B and then workload can be distributed between the two systems.

18. Defendant has had knowledge of the ’675 Patent and its infringement since at least the filing of the original Complaint in this action, or shortly thereafter, including by way of this lawsuit. By the time of trial, Defendant will have known and intended (since receiving such notice) that its continued actions would actively induce and contribute to the infringement of the claims of the ’675 Patent.

19. Defendant’s affirmative acts of making, using, selling, offering for sale, and/or importing the Accused Instrumentalities have induced and continue to induce users of the Accused Instrumentalities to use the Accused Instrumentalities in their normal and customary way to infringe the claims of the ’675 Patent. Use of the Accused Instrumentalities in their ordinary and customary fashion results in infringement of the claims of the ’675 Patent.

20. For example, Defendant explains to customers the benefits of using the Accused Instrumentalities, such as by touting their advantages of synchronizing settings among multiple devices. Defendant also induces its customers to use the Accused Instrumentalities to infringe other claims of the ’675 Patent. Defendant specifically intended and was aware that the normal and customary use of the Accused Instrumentalities on compatible systems would infringe the ’675 Patent. Defendant performed the acts that constitute induced infringement, and would induce actual infringement, with the

knowledge of the ’675 Patent and with the knowledge, or willful blindness to the probability, that the induced acts would constitute infringement. On information and belief, Defendant engaged in such inducement to promote the sales of the Accused Instrumentalities, e.g., through its user manuals, product support, marketing materials, demonstrations, installation support, and training materials to actively induce the users of the accused products to infringe the ’675 Patent. Accordingly, Defendant has induced and continues to induce end users of the accused products to use the accused products in their ordinary and customary way with compatible systems to make and/or use systems infringing the ’675 Patent, knowing that such use of the Accused Instrumentalities with compatible systems will result in infringement of the ’675 Patent. Accordingly, Defendant has been (since at least as of filing of the original complaint), and currently is, inducing infringement of the ’675 Patent, in violation of 35 U.S.C. § 271(b).

21. For similar reasons, Defendant also infringes the ’675 Patent by supplying or causing to be supplied in or from the United States all or a substantial portion of the components of the Accused Instrumentalities, where such components are uncombined in whole or in part, in such manner as to actively induce the combination of such components outside of the United States in a manner that would infringe the ’675 Patent if such combination occurred within the United States. For example, Defendant supplies or causes to be supplied in or from the United States all or a substantial portion of the hardware (e.g., separate Teradata servers) and software (e.g., Teradata Unity software) components of the Accused Instrumentalities in such a manner as to actively induce the combination of such components (e.g., by instructing users to combine multiple Teradata servers into an infringing system) outside of the United States

22. Defendant has also infringed, and continues to infringe, claims of the ’675 Patent by offering to commercially distribute, commercially distributing, making, and/or importing the Accused Instrumentalities, which are used in practicing the process, or using the systems, of the ’675 Patent, and constitute a material part of the invention. Defendant knows the components in the Accused Instrumentalities to be especially made or especially adapted for use in infringement of the ’674 Patent, not a staple article, and not a commodity of commerce suitable for substantial noninfringing use. For example, the ordinary way of using the Accused Instrumentalities infringes the patent claims, and as such, is especially adapted for use in infringement. Accordingly, Defendant has been, and currently is, contributorily infringing the ’675 Patent, in violation of 35 U.S.C. § 271(c).

23. Defendants also indirectly infringe the ’675 Patent by supplying or causing to be supplied in or from the United States components of the Accused Instrumentalities that are especially made or especially adapted for use in infringing the ’675 Patent and are not a staple article or commodity of commerce suitable for substantial non-infringing use, and where such components are uncombined in whole or in part, knowing that such components are so made or adapted and intending that such components are combined outside of the United States in a manner that would infringe the ’675 Patent if such combination occurred within the United States. Because the Accused Instrumentalities are designed to operate as the claimed system and apparatus, the Accused Instrumentalities have no substantial non-infringing uses, and any other uses would be unusual, far-fetched, illusory, impractical, occasional, aberrant, or experimental. For example, Defendant supplies or causes to be supplied in or from the United States all or a substantial portion of the hardware (e.g., separate Teradata servers) and software (e.g., Teradata Unity software)

components that are especially made or especially adapted for use in the Accused Instrumentalities, where such hardware and software components are not staple articles or commodities of commerce suitable for substantial noninfringing use, knowing that such components are so made or adapted and intending that such components are combined outside of the United States, as evidenced by Defendant's own actions or instructions to users in, e.g., combining multiple Teradata servers into infringing systems, and enabling and configuring the infringing functionalities of the Accused Instrumentalities.

24. As a result of Defendant's infringement of the '675 Patent, Plaintiff Data Scape is entitled to monetary damages in an amount adequate to compensate for each Defendant's infringement, but in no event less than a reasonable royalty for the use made of the invention by each Defendant, together with interest and costs as fixed by the Court.

COUNT II

INFRINGEMENT OF U.S. PATENT NO. 10,027,751

25. Data Scape is the owner by assignment of United States Patent No. 10,027,751 ("the '751 Patent"), entitled "Communication System And Its Method and Communication Apparatus And Its Method." The '751 Patent was duly and legally issued by the United States Patent and Trademark Office on July 17, 2018. A true and correct copy of the '751 Patent is included as Exhibit B.

26. Defendant has offered for sale, sold and/or imported into the United States products and services that infringe the '751 patent, and continues to do so. By way of illustrative example, these infringing products and services include, without limitation, Defendant's products and services, *e.g.*, Unity, Unity Director, Unity Loader, Data Mover, Teradata Managed Server Unity, Unity Director/Loader Model 8-81X and Expansion

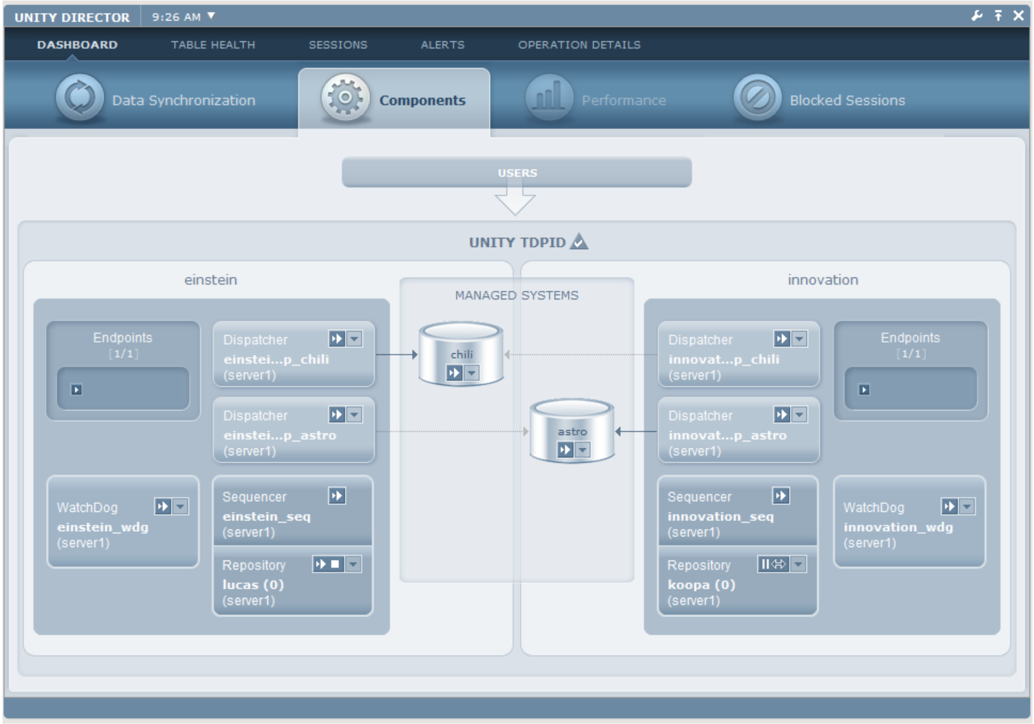
Server Model 8-81XE, Unity Source Link Server, Teradata Managed Server Unity Data Mover, and all versions and variations thereof since the issuance of the ’751 Patent (“Accused Instrumentalities”).

27. Defendant has directly infringed and continues to infringe the ’751 Patent, for example, by making, selling, offering for sale, and/or importing the Accused Instrumentalities, and through its own use and testing of the Accused Instrumentalities. Defendant uses the Accused Instrumentalities for its own internal non-testing business purposes, while testing the Accused Instrumentalities, and while providing technical support and repair services for the Accused Instrumentalities to its customers.

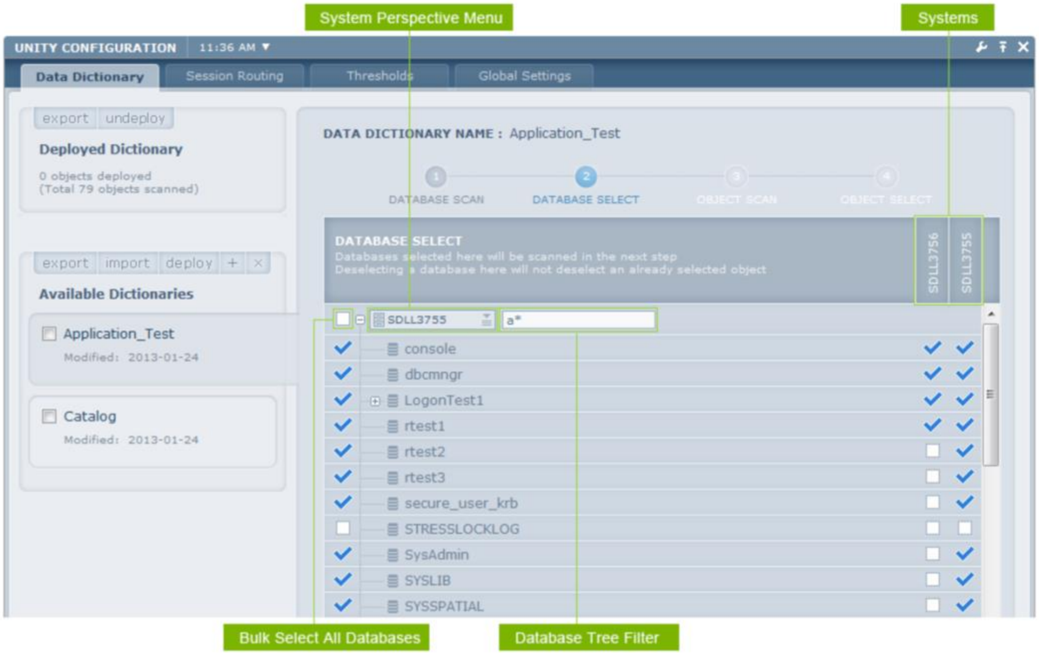
28. For example, the Accused Instrumentalities infringe Claim 1 (as well as other claims) of the ’751 Patent. One non-limiting example of the Accused Instrumentalities’ infringement is presented below:

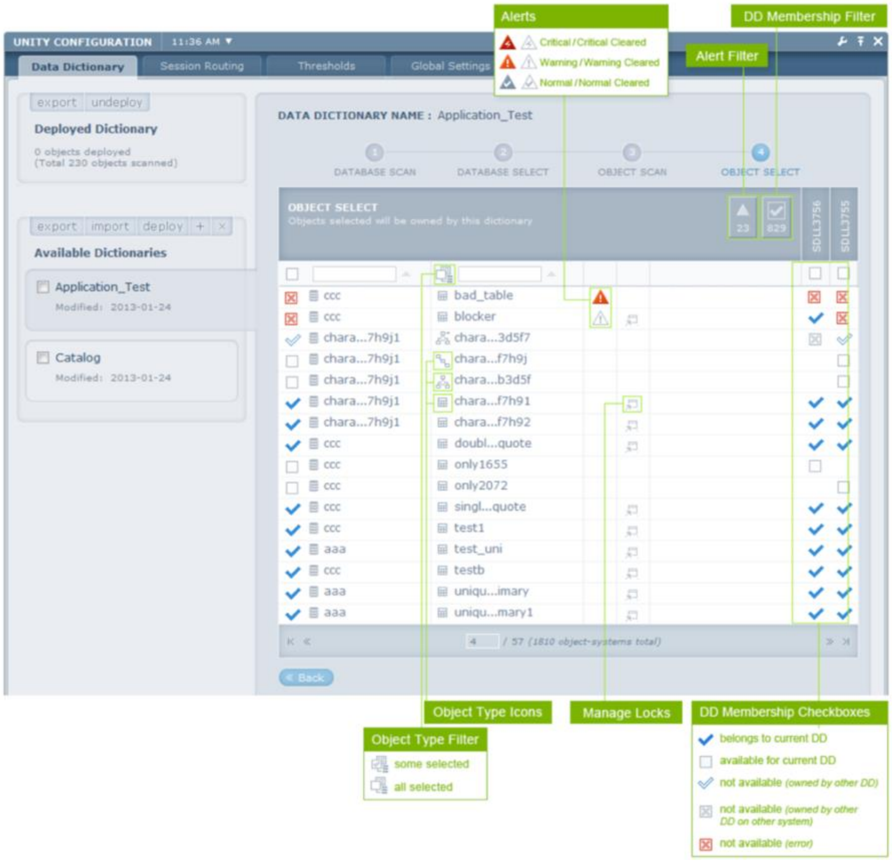
29. The Accused instrumentalities include “[a] communication apparatus configured to transmit data to an apparatus.” For example, the Accused Instrumentalities communicate data stored on one device (e.g. a device running Unity, Unity Director, Unity Loader, Teradata Bulk Transfer, Teradata Data Mover) to another device with a storage medium (e.g. a Teradata Database system). See, e.g., Teradata Unity Datasheet (EB7192.pdf), available at http://assets.teradata.com/resourceCenter/downloads/Datasheets/EB7192.pdf. (“Another benefit of Teradata Unity is the capability to synchronize multiple Teradata systems. The SQL Multicast feature delivers SQL commands to all participating systems in the Teradata Analytical Ecosystem. Teradata Unity will automatically queue up and dispatch the incoming SQL commands in the order in which they’re received, maintaining consistency

and integrity across systems."); http://downloads.teradata.com/uda/articles/unity-director-initial-configuration-and-setup:


UNITY DIRECTOR
9:26 AM ▼
DASHBOARD
TABLE HEALTH
SESSIONS
ALERTS
OPERATION DETAILS
Data Synchronization
Components
Performance
Blocked Sessions
USERS
UNITY TDPID
einstein
innovation
MANAGED SYSTEMS
Endpoints [1/1]
Dispatcher einstei...p_chili (server1)
Dispatcher einstei...p_astro (server1)
chili
astro
Dispatcher innovat...p_chili (server1)
Dispatcher innovat...p_astro (server1)
Endpoints [1/1]
WatchDog einstein_wdg (server1)
Sequencer einstein_seq (server1)
Repository lucas (0) (server1)
Sequencer innovation_seq (server1)
Repository koopa (0) (server1)
WatchDog innovation_wdg (server1)

b. Select databases used in the application:





UNITY CONFIGURATION 11:36 AM ▼

Data Dictionary | Session Routing | Thresholds | Global Settings

export | undeploy

Deployed Dictionary

0 objects deployed
(Total 230 objects scanned)

export | import | deploy | + | ×

Available Dictionaries

□ Application_Test
Modified: 2013-01-24

□ Catalog
Modified: 2013-01-24

DATA DICTIONARY NAME : Application_Test

1 DATABASE SCAN — 2 DATABASE SELECT — 3 OBJECT SCAN — 4 OBJECT SELECT

OBJECT SELECT
Objects selected will be owned by this dictionary

▲ 23 | ☑ 829

□					SDLL3756 □	SDLL3755 □
☒	ccc	bad_table	▲		☒	☒
☒	ccc	blocker	△	🔒	✓	☒
✓	chara...7h9j1	chara...3d5f7			☒	✓
□	chara...7h9j1	chara...f7h9j				□
□	chara...7h9j1	chara...b3d5f				□
✓	chara...7h9j1	chara...f7h91		🔒	✓	✓
✓	chara...7h9j1	chara...f7h92		🔒	✓	✓
✓	ccc	doubl...quote		🔒	✓	✓
□	ccc	only1655			□	
□	ccc	only2072				□
✓	ccc	singl...quote		🔒	✓	✓
✓	ccc	test1		🔒	✓	✓
✓	aaa	test_uni		🔒	✓	✓
✓	ccc	testb		🔒	✓	✓
✓	aaa	uniqu...imary		🔒	✓	✓
✓	aaa	uniqu...mary1		🔒	✓	✓

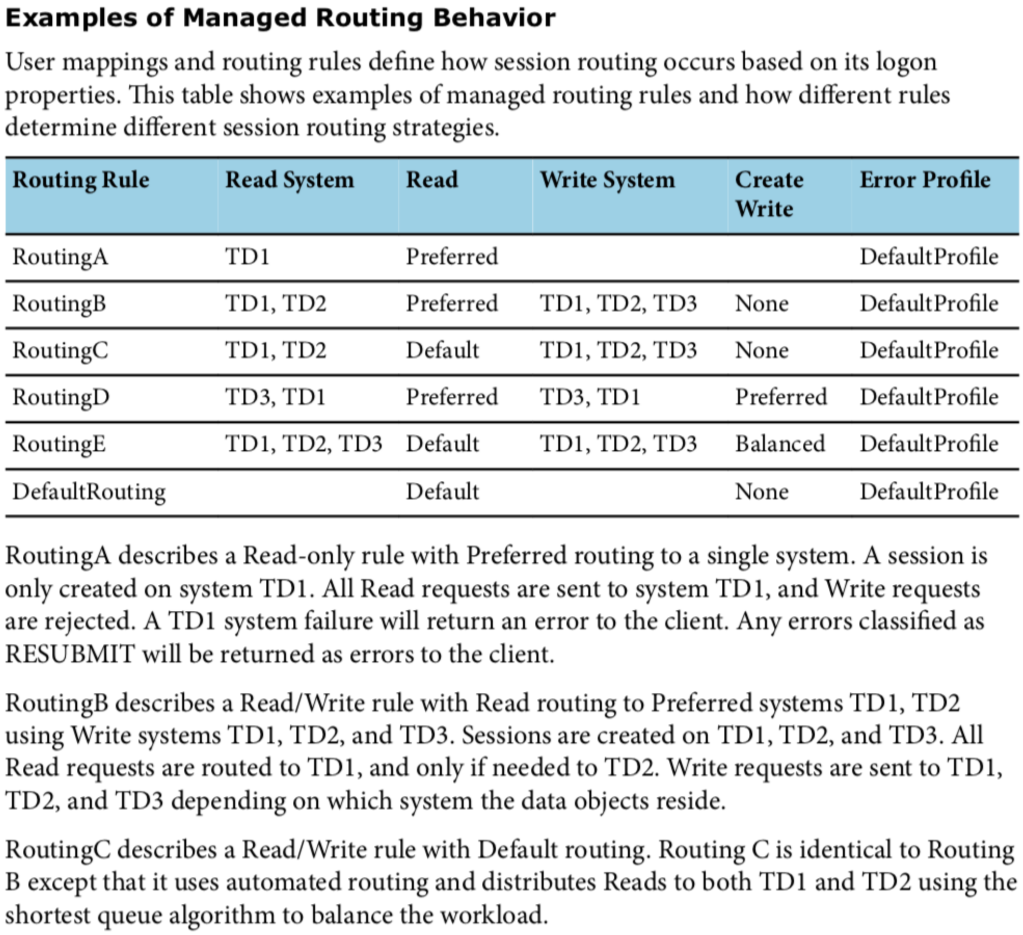
4 / 57 (1810 object-systems total)

« Back

Alerts
▲ △ Critical / Critical Cleared
▲ △ Warning / Warning Cleared
▲ △ Normal / Normal Cleared
DD Membership Filter
Alert Filter
Object Type Icons
Manage Locks
DD Membership Checkboxes
✓ belongs to current DD
□ available for current DD
✓ not available (owned by other DD)
☒ not available (owned by other DD on other system)
☒ not available (error)
Object Type Filter
some selected
all selected

30. The Accused instrumentalities include a communication apparatus comprising “a hardware storage medium configured to store management information of data to be transferred to the apparatus.” For example, the Accused Instrumentalities include disk drives and/or solid state disks, which are necessary for the operation of the Accused Instrumentalities. *See, e.g.*, Teradata Managed Server for Cabinet and Server Models 4X5, 4X7, and 6X7 Product and Site Preparation Guide, *available at* https://docs.teradata.com/reader/DdAmXXaBagDQwtH6uaApUg/ce~Xolu7GagxHN5mOr4eRQ (“The following table outlines the features of the Teradata Managed Server 687 Unity models. *** Disk drives: Six 3.5-inch 450 GB or 600 GB SAS 15K RPM hard drives”); Teradata Unity User Guide, Release 15.10 (2520-066K.pdf) (“Unity supports different sized Teradata systems that load at different rates and minimizes impact on the Bulk Load client when a system is offline. Unity manages an internal disk array to

accommodate the disk space requirements for landing data before loading the data to Teradata systems. Data is landed to disk for any of the following reasons: • Data cannot be loaded to one or more of the target systems due to target, error, or restart tables in the Interrupted state or an unavailable target system. • A data checkpoint did not occur on all systems, and there is insufficient memory to store the data in memory. • A data checkpoint is successful on one system, but not on the others. Unity deletes landed data stored on the disk array after the data successfully loads to the Teradata systems, and the begin loading record of the Bulk Load job is overwritten in the Recovery Log. Management of the disk space by Unity reduces the total amount of disk space required for the load files."). As further example, the Accused Instrumentalities store information about data to be transferred to another device:



Examples of Managed Routing Behavior

User mappings and routing rules define how session routing occurs based on its logon properties. This table shows examples of managed routing rules and how different rules determine different session routing strategies.

Routing Rule	Read System	Read	Write System	Create Write	Error Profile
RoutingA	TD1	Preferred			DefaultProfile
RoutingB	TD1, TD2	Preferred	TD1, TD2, TD3	None	DefaultProfile
RoutingC	TD1, TD2	Default	TD1, TD2, TD3	None	DefaultProfile
RoutingD	TD3, TD1	Preferred	TD3, TD1	Preferred	DefaultProfile
RoutingE	TD1, TD2, TD3	Default	TD1, TD2, TD3	Balanced	DefaultProfile
DefaultRouting		Default		None	DefaultProfile

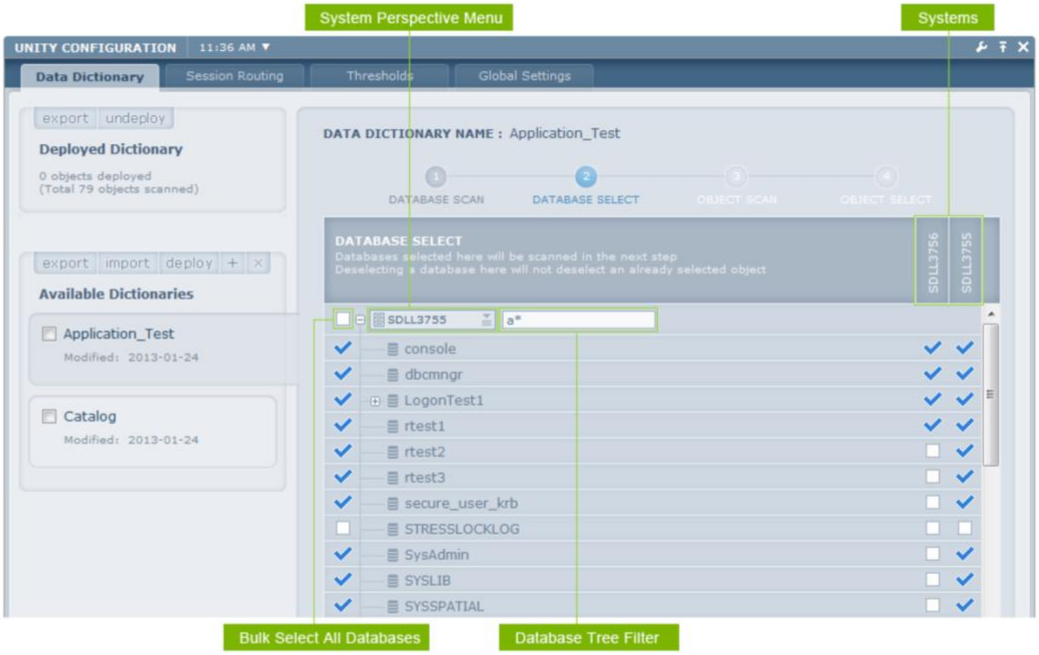
RoutingA describes a Read-only rule with Preferred routing to a single system. A session is only created on system TD1. All Read requests are sent to system TD1, and Write requests are rejected. A TD1 system failure will return an error to the client. Any errors classified as RESUBMIT will be returned as errors to the client.

RoutingB describes a Read/Write rule with Read routing to Preferred systems TD1, TD2 using Write systems TD1, TD2, and TD3. Sessions are created on TD1, TD2, and TD3. All Read requests are routed to TD1, and only if needed to TD2. Write requests are sent to TD1, TD2, and TD3 depending on which system the data objects reside.

RoutingC describes a Read/Write rule with Default routing. Routing C is identical to Routing B except that it uses automated routing and distributes Reads to both TD1 and TD2 using the shortest queue algorithm to balance the workload.

See also http://downloads.teradata.com/uda/articles/unity-director-initial-configuration-and-setup:

b. Select databases used in the application:



Deployed Dictionary
0 objects deployed
(Total 79 objects scanned)

Available Dictionaries

☐ Application_Test
Modified: 2013-01-24

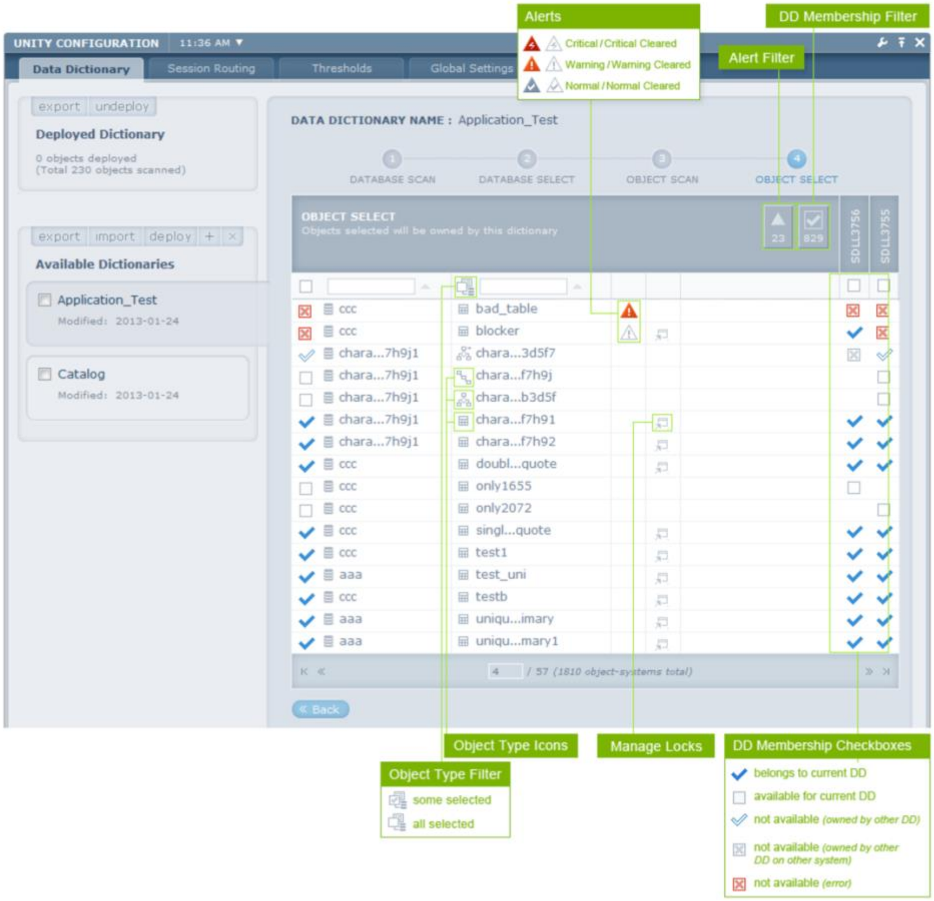
☐ Catalog
Modified: 2013-01-24

DATA DICTIONARY NAME : Application_Test

1 DATABASE SCAN — 2 DATABASE SELECT — 3 OBJECT SCAN — 4 OBJECT SELECT

DATABASE SELECT
Databases selected here will be scanned in the next step
Deselecting a database here will not deselect an already selected object

☐	SDLL3755	a*	SDLL3756	SDLL3755
✓	console		✓	✓
✓	dbcmngr		✓	✓
✓	LogonTest1		✓	✓
✓	rtest1		✓	✓
✓	rtest2		☐	✓
✓	rtest3		☐	✓
✓	secure_user_krb		☐	✓
☐	STRESSLOCKLOG		☐	☐
✓	SysAdmin		☐	✓
✓	SYSLIB		☐	✓
✓	SYSSPATIAL		☐	✓



Deployed Dictionary
0 objects deployed
(Total 230 objects scanned)

Available Dictionaries

☐ Application_Test
Modified: 2013-01-24

☐ Catalog
Modified: 2013-01-24

DATA DICTIONARY NAME : Application_Test

1 DATABASE SCAN — 2 DATABASE SELECT — 3 OBJECT SCAN — 4 OBJECT SELECT

OBJECT SELECT
Objects selected will be owned by this dictionary

☐					SDLL3756	SDLL3755
☒	ccc	bad_table	⚠		☒	☒
☒	ccc	blocker	⚠	⧉	✓	☒
✓	chara...7h9j1	chara...3d5f7			☒	✓
☐	chara...7h9j1	chara...f7h9j				☐
☐	chara...7h9j1	chara...b3d5f				☐
✓	chara...7h9j1	chara...f7h91		⧉	✓	✓
✓	chara...7h9j1	chara...f7h92		⧉	✓	✓
✓	ccc	doubl...quote		⧉	✓	✓
☐	ccc	only1655			☐	
☐	ccc	only2072				☐
✓	ccc	singl...quote		⧉	✓	✓
✓	ccc	test1		⧉	✓	✓
✓	aaa	test_uni		⧉	✓	✓
✓	ccc	testb		⧉	✓	✓
✓	aaa	uniqu...imary		⧉	✓	✓
✓	aaa	uniqu...mary1		⧉	✓	✓

K « 4 / 57 (1810 object-systems total) » ›I

DD Membership Checkboxes
- ✓ belongs to current DD
- ☐ available for current DD
- ✓ not available *(owned by other DD)*
- ☒ not available *(owned by other DD on other system)*
- ☒ not available *(error)*

31. The Accused instrumentalities include a communication apparatus comprising “a communicator configured to communicate data with the apparatus.” For example, the Accused Instrumentalities communicate with external Teradata database servers. See, e.g., EB7192.pdf ("Another benefit of Teradata Unity is the capability to synchronize multiple Teradata systems. The SQL Multicast feature delivers SQL commands to all participating systems in the Teradata Analytical Ecosystem. Teradata Unity will automatically queue up and dispatch the incoming SQL commands in the order in which they're received, maintaining consistency and integrity across systems."); *id*:

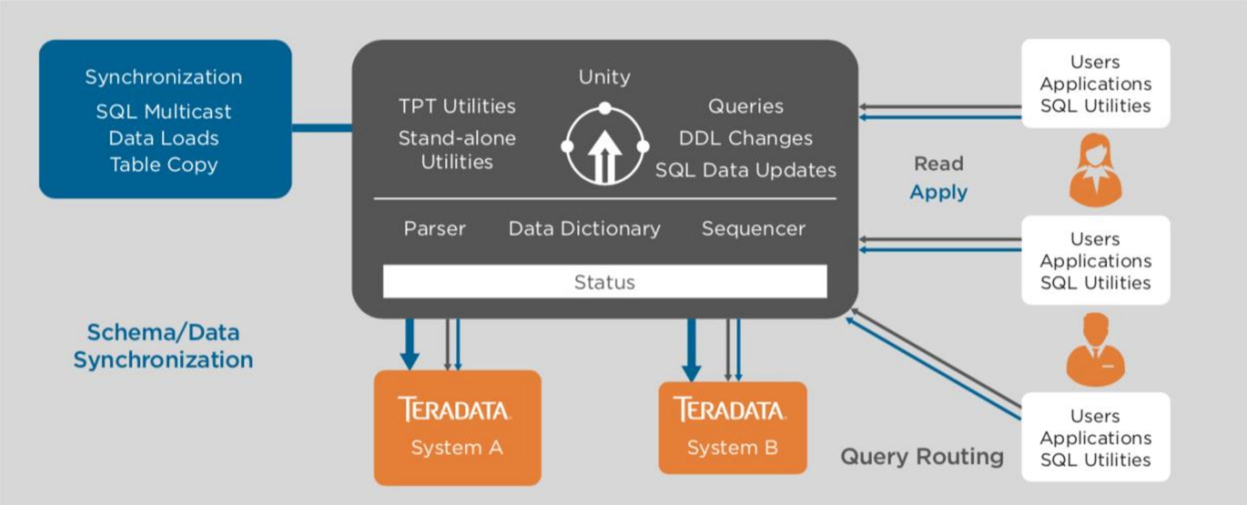
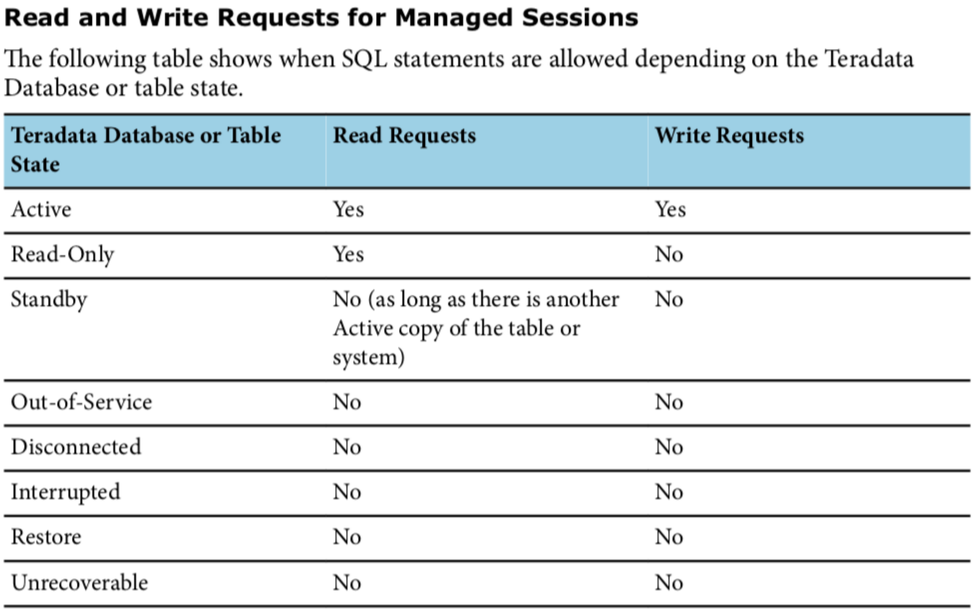


Figure 1. Unity—Query Routing and Data Synchronization.

32. The Accused instrumentalities include a communication apparatus comprising “a detector configured to detect whether the communication apparatus and the apparatus are connected.” For example, the Accused Instrumentalities detects the connection state with the Teradata Database and transfers data only when the database is in “Active” or “Read-Only” state, i.e. connected, and not when the Teradata Database is in “Disconnected” state. See, e.g., 2520-066K.pdf:



Read and Write Requests for Managed Sessions

The following table shows when SQL statements are allowed depending on the Teradata Database or table state.

Teradata Database or Table State	Read Requests	Write Requests
Active	Yes	Yes
Read-Only	Yes	No
Standby	No (as long as there is another Active copy of the table or system)	No
Out-of-Service	No	No
Disconnected	No	No
Interrupted	No	No
Restore	No	No
Unrecoverable	No	No

33. The Accused instrumentalities include a communication apparatus comprising "an editor configured to select certain data to be transferred and to edit the management information based on the selection without regard to the connection of the communication apparatus and the apparatus." For example, the Accused Instrumentalities include an editor to select data to send to individual Teradata database systems based on, *e.g.*, managed routing rules, and edits management information based on the selection. This editing is done without regard to the connection between the Accused Instrumentalities and the target database system, *e.g.*, when the Accused Instrumentalities receive commands to create or drop tables using SQL Data Definition Language. *See, e.g.*, 2520-066K.pdf:

Examples of Managed Routing Behavior

User mappings and routing rules define how session routing occurs based on its logon properties. This table shows examples of managed routing rules and how different rules determine different session routing strategies.

Routing Rule	Read System	Read	Write System	Create Write	Error Profile
RoutingA	TD1	Preferred			DefaultProfile
RoutingB	TD1, TD2	Preferred	TD1, TD2, TD3	None	DefaultProfile
RoutingC	TD1, TD2	Default	TD1, TD2, TD3	None	DefaultProfile
RoutingD	TD3, TD1	Preferred	TD3, TD1	Preferred	DefaultProfile
RoutingE	TD1, TD2, TD3	Default	TD1, TD2, TD3	Balanced	DefaultProfile
DefaultRouting		Default		None	DefaultProfile

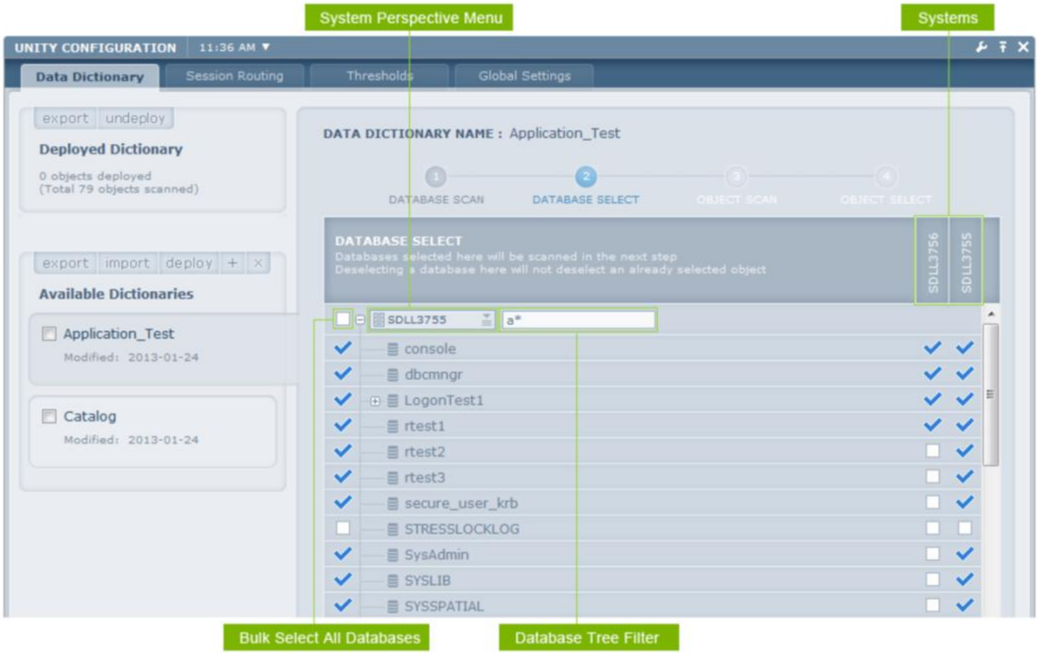
RoutingA describes a Read-only rule with Preferred routing to a single system. A session is only created on system TD1. All Read requests are sent to system TD1, and Write requests are rejected. A TD1 system failure will return an error to the client. Any errors classified as RESUBMIT will be returned as errors to the client.

RoutingB describes a Read/Write rule with Read routing to Preferred systems TD1, TD2 using Write systems TD1, TD2, and TD3. Sessions are created on TD1, TD2, and TD3. All Read requests are routed to TD1, and only if needed to TD2. Write requests are sent to TD1, TD2, and TD3 depending on which system the data objects reside.

RoutingC describes a Read/Write rule with Default routing. Routing C is identical to Routing B except that it uses automated routing and distributes Reads to both TD1 and TD2 using the shortest queue algorithm to balance the workload.

See also http://downloads.teradata.com/uda/articles/unity-director-initial-configuration-and-setup:

b. Select databases used in the application:



0 objects deployed
(Total 79 objects scanned)

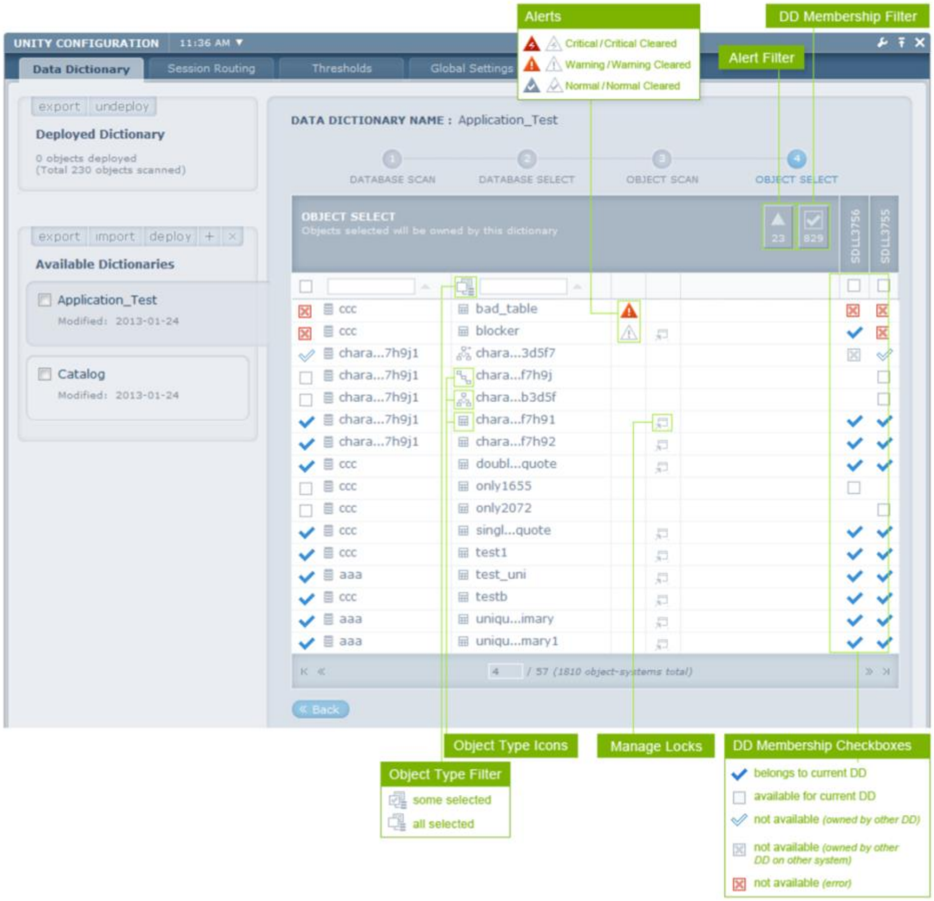
☐ Application_Test
Modified: 2013-01-24

☐ Catalog
Modified: 2013-01-24

DATA DICTIONARY NAME : Application_Test

DATABASE SELECT
Databases selected here will be scanned in the next step
Deselecting a database here will not deselect an already selected object

☐	SDLL3755	SDLL3756	SDLL3755
	a*		
✓	console	✓	✓
✓	dbcmngr	✓	✓
✓	LogonTest1	✓	✓
✓	rtest1	✓	✓
✓	rtest2	☐	✓
✓	rtest3	☐	✓
✓	secure_user_krb	☐	✓
☐	STRESSLOCKLOG	☐	☐
✓	SysAdmin	☐	✓
✓	SYSLIB	☐	✓
✓	SYSSPATIAL	☐	✓



Deployed Dictionary

0 objects deployed
(Total 230 objects scanned)

Available Dictionaries

□ Application_Test
Modified: 2013-01-24

□ Catalog
Modified: 2013-01-24

DATA DICTIONARY NAME : Application_Test

1 DATABASE SCAN — 2 DATABASE SELECT — 3 OBJECT SCAN — 4 OBJECT SELECT

OBJECT SELECT
Objects selected will be owned by this dictionary

▲ 23 ☑ 829

□					SDLL3756 □	SDLL3755 □
☒	ccc	bad_table	⚠		☒	☒
☒	ccc	blocker	⚠	🔒	✓	☒
✓	chara...7h9j1	chara...3d5f7			☒	✓
□	chara...7h9j1	chara...f7h9j				□
□	chara...7h9j1	chara...b3d5f				□
✓	chara...7h9j1	chara...f7h91		🔒	✓	✓
✓	chara...7h9j1	chara...f7h92		🔒	✓	✓
✓	ccc	doubl...quote		🔒	✓	✓
□	ccc	only1655			□	
□	ccc	only2072				□
✓	ccc	singl...quote		🔒	✓	✓
✓	ccc	test1		🔒	✓	✓
✓	aaa	test_uni		🔒	✓	✓
✓	ccc	testb		🔒	✓	✓
✓	aaa	uniqu...imary		🔒	✓	✓
✓	aaa	uniqu...mary1		🔒	✓	✓

4 / 57 (1810 object-systems total)

✓ belongs to current DD
□ available for current DD
✓ not available *(owned by other DD)*
☒ not available *(owned by other DD on other system)*
☒ not available *(error)*

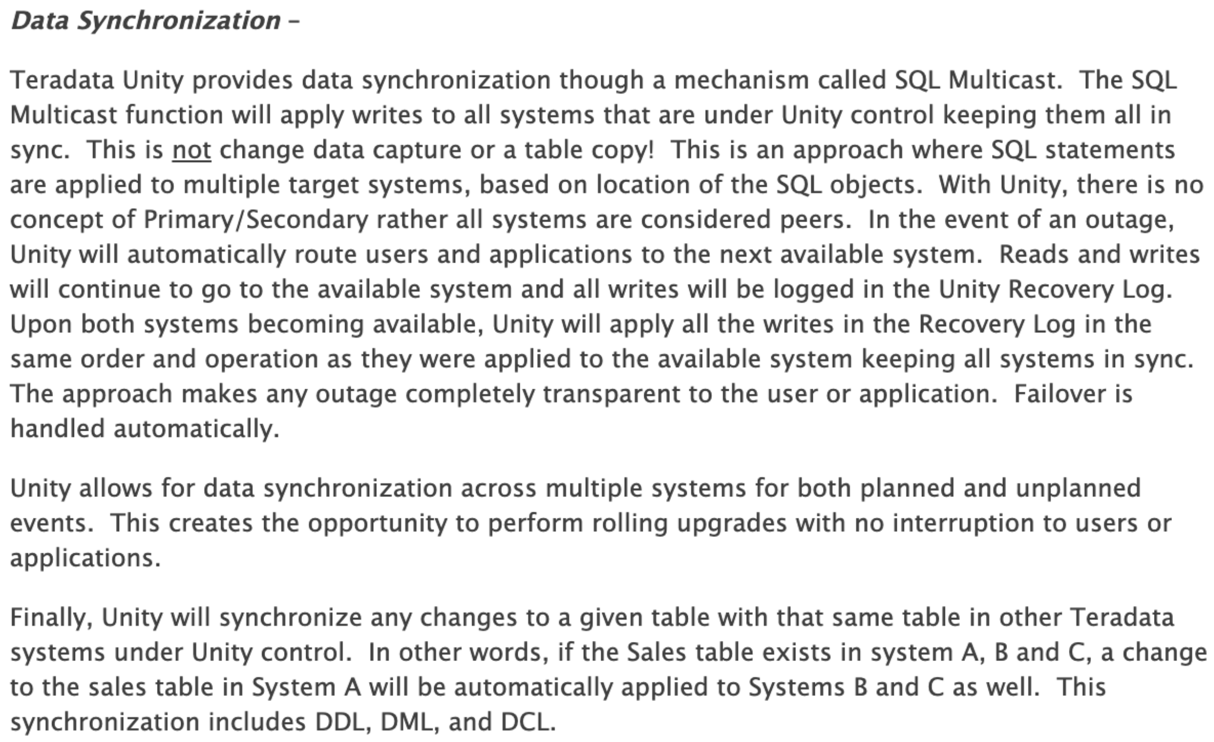
See also http://developer.teradata.com/uda/articles/unity-13-10-foundation-catalog-deploy-examples:

Dynamic catalog creation

Dynamically deploying additional objects for Teradata Unity to manage into an existing user or database can be accomplished by executing DDL directly through Teradata Unity. In the final video clip I will run a bteq script which creates hundreds of tables, views, and macros. Most likely you will want to use your existing ant build scripts to dynamically create objects as you migrate from your development and test environments to production.

> **Note**: Any DDL change requires a Dictionary rescan, *unless* the change is one of the following:
>
> - create/drop table
> - create/drop view
> - create/drop macro

See also https://developer.teradata.com/uda/articles/stay-in-sync-introducing-teradata-unity-13-10:



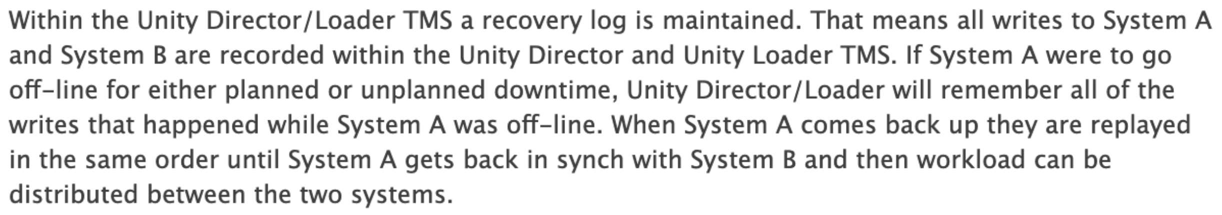
Data Synchronization –

Teradata Unity provides data synchronization though a mechanism called SQL Multicast. The SQL Multicast function will apply writes to all systems that are under Unity control keeping them all in sync. This is not change data capture or a table copy! This is an approach where SQL statements are applied to multiple target systems, based on location of the SQL objects. With Unity, there is no concept of Primary/Secondary rather all systems are considered peers. In the event of an outage, Unity will automatically route users and applications to the next available system. Reads and writes will continue to go to the available system and all writes will be logged in the Unity Recovery Log. Upon both systems becoming available, Unity will apply all the writes in the Recovery Log in the same order and operation as they were applied to the available system keeping all systems in sync. The approach makes any outage completely transparent to the user or application. Failover is handled automatically.

Unity allows for data synchronization across multiple systems for both planned and unplanned events. This creates the opportunity to perform rolling upgrades with no interruption to users or applications.

Finally, Unity will synchronize any changes to a given table with that same table in other Teradata systems under Unity control. In other words, if the Sales table exists in system A, B and C, a change to the sales table in System A will be automatically applied to Systems B and C as well. This synchronization includes DDL, DML, and DCL.

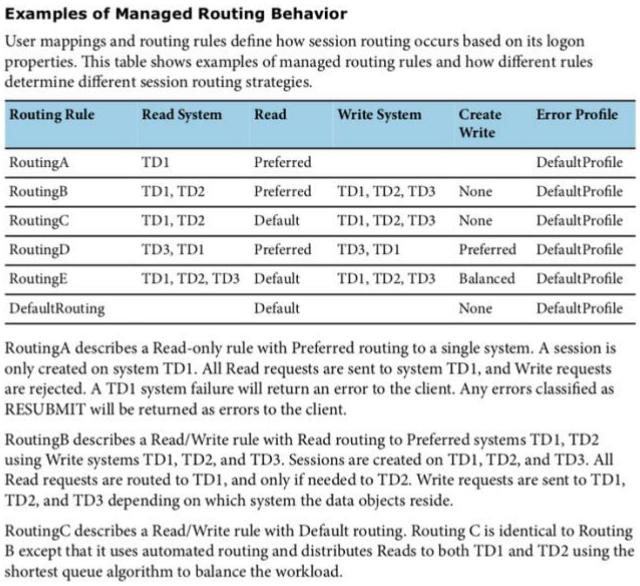
See also, e.g., https://developer.teradata.com/general/articles/multi-active-systems-with-new-unity-directorloader-14-11:



Within the Unity Director/Loader TMS a recovery log is maintained. That means all writes to System A and System B are recorded within the Unity Director and Unity Loader TMS. If System A were to go off-line for either planned or unplanned downtime, Unity Director/Loader will remember all of the writes that happened while System A was off-line. When System A comes back up they are replayed in the same order until System A gets back in synch with System B and then workload can be distributed between the two systems.

34. The Accused instrumentalities include a communication apparatus comprising "a controller configured to control transfer of the selected data stored in the communication apparatus to the apparatus via the communicator based on the management information edited by the editor when the detector detects that the communication apparatus and the apparatus are connected." For example, the Accused Instrumentalities transmits data to the Teradata Database server to an external database system that is

identified, when it is connected by a unique identifier called a TDPID, and the Accused Instrumentalities controls what data to transmit based on, e.g., managed routing rules, and edits management information based on the selection. See, e.g., 2520-066K.pdf (“DICTIONARY SET PREFERRED: Purpose: Sets the Preferred Write system to the Teradata Database system identified by the TDPID value. To set the Preferred Write system for all tables in a database, specify only the name of the database. To set the Preferred Write system for a single table, specify both the database name and the table name. *** tdpid: Unique identifier (TDPID) of a Teradata Database system.”); EB7192.pdf (“Taken together, the data loading and query management work as one to sequence database changes and query requests. This ensures that data updates and database structure changes are always applied in the order received; automatically coordinating and maintaining order and consistency across systems. And, it is specifically designed to work with systems that are not identical. For example, if the primary integrated data warehouse contains 100 percent of the data, and there is a second system for high availability that holds a 30 percent subset of that data, Teradata Unity understands the capabilities and limitations of each system and will route data loads accordingly.”); *id:*



Examples of Managed Routing Behavior

User mappings and routing rules define how session routing occurs based on its logon properties. This table shows examples of managed routing rules and how different rules determine different session routing strategies.

Routing Rule	Read System	Read	Write System	Create Write	Error Profile
RoutingA	TD1	Preferred			DefaultProfile
RoutingB	TD1, TD2	Preferred	TD1, TD2, TD3	None	DefaultProfile
RoutingC	TD1, TD2	Default	TD1, TD2, TD3	None	DefaultProfile
RoutingD	TD3, TD1	Preferred	TD3, TD1	Preferred	DefaultProfile
RoutingE	TD1, TD2, TD3	Default	TD1, TD2, TD3	Balanced	DefaultProfile
DefaultRouting		Default		None	DefaultProfile

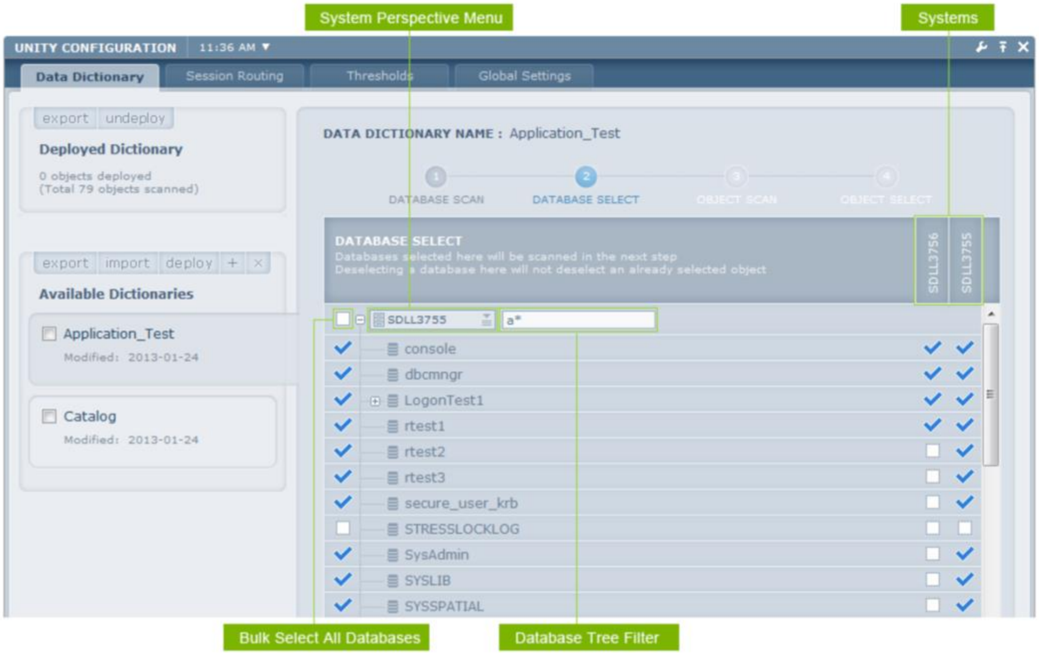
RoutingA describes a Read-only rule with Preferred routing to a single system. A session is only created on system TD1. All Read requests are sent to system TD1, and Write requests are rejected. A TD1 system failure will return an error to the client. Any errors classified as RESUBMIT will be returned as errors to the client.

RoutingB describes a Read/Write rule with Read routing to Preferred systems TD1, TD2 using Write systems TD1, TD2, and TD3. Sessions are created on TD1, TD2, and TD3. All Read requests are routed to TD1, and only if needed to TD2. Write requests are sent to TD1, TD2, and TD3 depending on which system the data objects reside.

RoutingC describes a Read/Write rule with Default routing. Routing C is identical to Routing B except that it uses automated routing and distributes Reads to both TD1 and TD2 using the shortest queue algorithm to balance the workload.

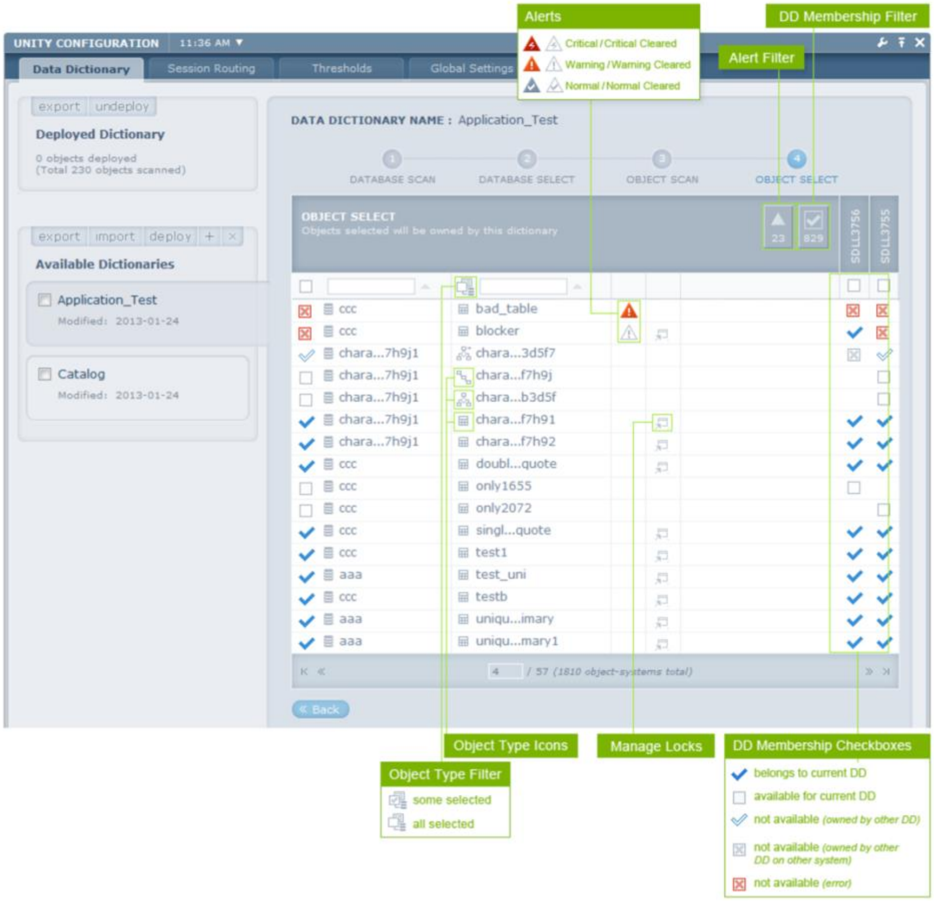
See also http://downloads.teradata.com/uda/articles/unity-director-initial-configuration-and-setup:

b. Select databases used in the application:



DATABASE SELECT
Databases selected here will be scanned in the next step
Deselecting a database here will not deselect an already selected object

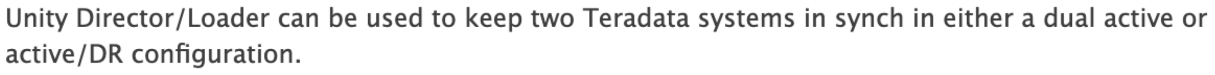
		SDLL3756	SDLL3755
✓	console	✓	✓
✓	dbcmngr	✓	✓
✓	LogonTest1	✓	✓
✓	rtest1	✓	✓
✓	rtest2	□	✓
✓	rtest3	□	✓
✓	secure_user_krb	□	✓
□	STRESSLOCKLOG	□	□
✓	SysAdmin	□	✓
✓	SYSLIB	□	✓
✓	SYSSPATIAL	□	✓



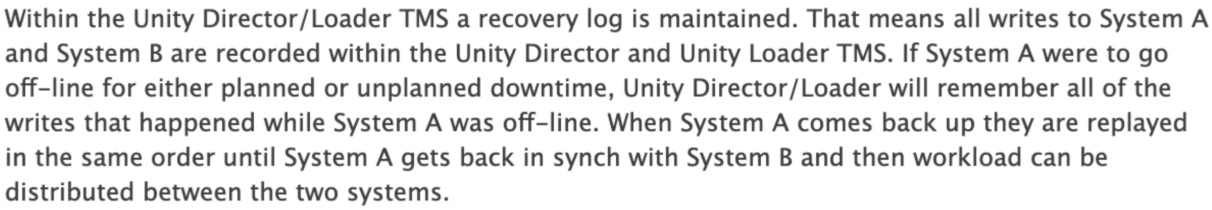
OBJECT SELECT
Objects selected will be owned by this dictionary

			SDLL3756	SDLL3755
☒	ccc	bad_table	☒	☒
☒	ccc	blocker	✓	☒
✓	chara...7h9j1	chara...3d5f7	☒	✓
□	chara...7h9j1	chara...f7h9j		□
□	chara...7h9j1	chara...b3d5f		□
✓	chara...7h9j1	chara...f7h91	✓	✓
✓	chara...7h9j1	chara...f7h92	✓	✓
✓	ccc	doubl...quote	✓	✓
□	ccc	only1655	□	
□	ccc	only2072		□
✓	ccc	singl...quote	✓	✓
✓	ccc	test1	✓	✓
✓	aaa	test_uni	✓	✓
✓	ccc	testb	✓	✓
✓	aaa	uniqu...imary	✓	✓
✓	aaa	uniqu...mary1	✓	✓

See also, e.g., https://developer.teradata.com/general/articles/multi-active-systems-with-new-unity-directorloader-14-11:

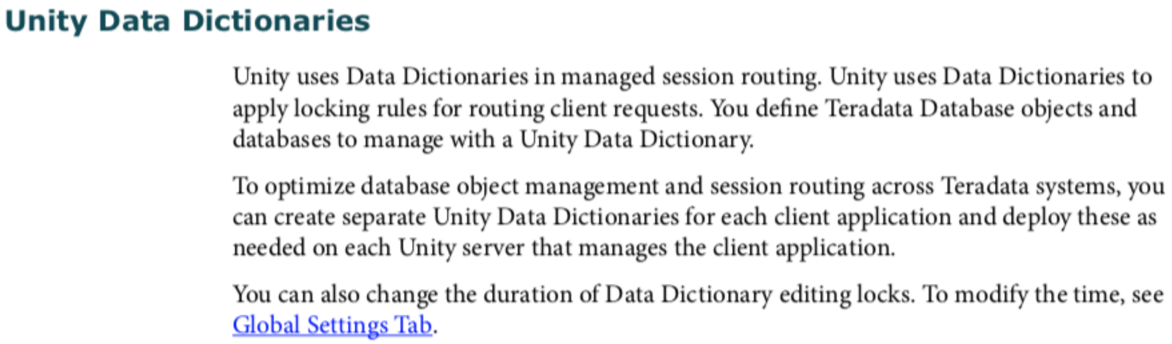


Unity Director/Loader can be used to keep two Teradata systems in synch in either a dual active or active/DR configuration.



Within the Unity Director/Loader TMS a recovery log is maintained. That means all writes to System A and System B are recorded within the Unity Director and Unity Loader TMS. If System A were to go off-line for either planned or unplanned downtime, Unity Director/Loader will remember all of the writes that happened while System A was off-line. When System A comes back up they are replayed in the same order until System A gets back in synch with System B and then workload can be distributed between the two systems.

35. The Accused instrumentalities include a communication apparatus comprising a controller configured to "compare the management information edited by the editor with management information of data stored in the apparatus." For example, the Accused Instrumentalities provide a mechanism to manage 'Data Dictionaries' that define what data/tables should be synced to a given database server. See, e.g., Teradata Unity User Guide, Release 15.10 (2520-066K.pdf):



Unity Data Dictionaries

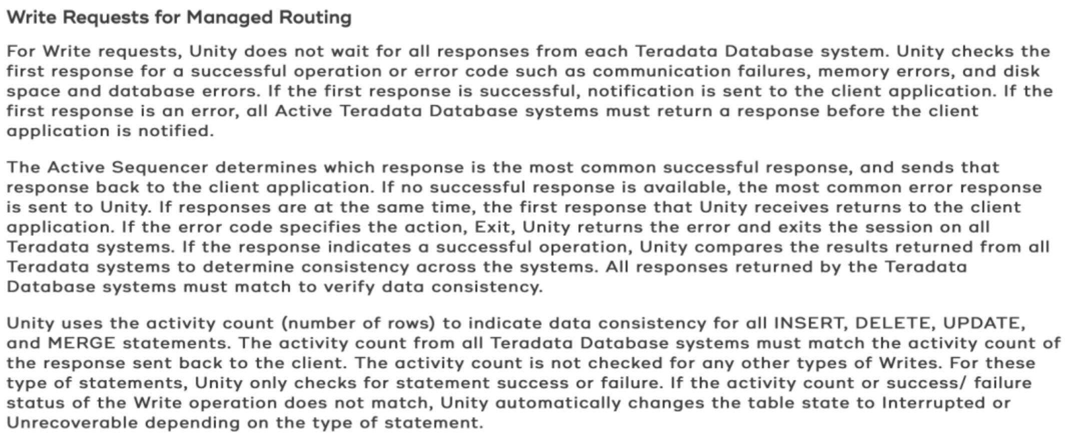
Unity uses Data Dictionaries in managed session routing. Unity uses Data Dictionaries to apply locking rules for routing client requests. You define Teradata Database objects and databases to manage with a Unity Data Dictionary.

To optimize database object management and session routing across Teradata systems, you can create separate Unity Data Dictionaries for each client application and deploy these as needed on each Unity server that manages the client application.

You can also change the duration of Data Dictionary editing locks. To modify the time, see Global Settings Tab.

36. The Accused instrumentalities include a communication apparatus comprising a controller configured to "determine a size of the selected data in the communication apparatus." For example, the Accused Instrumentalities will make certain there is the destination has the capacity to receive the size of the data to be transmitted.

See, *e.g.*, Teradata Managed Server for Cabinet and Server Models 4X5, 4X7, and 6X7 Product and Site Preparation Guide, *available at* https://docs.teradata.com/reader/DdAmXXaBagDQwtH6uaApUg/ce~Xolu7GagxHN5mOr4eRQ ("The following table outlines the features of the Teradata Managed Server 687 Unity models. *** Disk drives: Six 3.5-inch 450 GB or 600 GB SAS 15K RPM hard drives"); Teradata Unity User Guide, Release 15.10 (2520-066K.pdf) ("Unity supports different sized Teradata systems that load at different rates and minimizes impact on the Bulk Load client when a system is offline. Unity manages an internal disk array to accommodate the disk space requirements for landing data before loading the data to Teradata systems. Data is landed to disk for any of the following reasons: • Data cannot be loaded to one or more of the target systems due to target, error, or restart tables in the Interrupted state or an unavailable target system. • A data checkpoint did not occur on all systems, and there is insufficient memory to store the data in memory. • A data checkpoint is successful on one system, but not on the others. Unity deletes landed data stored on the disk array after the data successfully loads to the Teradata systems, and the begin loading record of the Bulk Load job is overwritten in the Recovery Log. Management of the disk space by Unity reduces the total amount of disk space required for the load files."). *See also* https://docs.teradata.com/reader/uQwTC22rxdNQl1IRU8Akcg/GNGC0tLo7fxkwXeczgnwLg:



Write Requests for Managed Routing

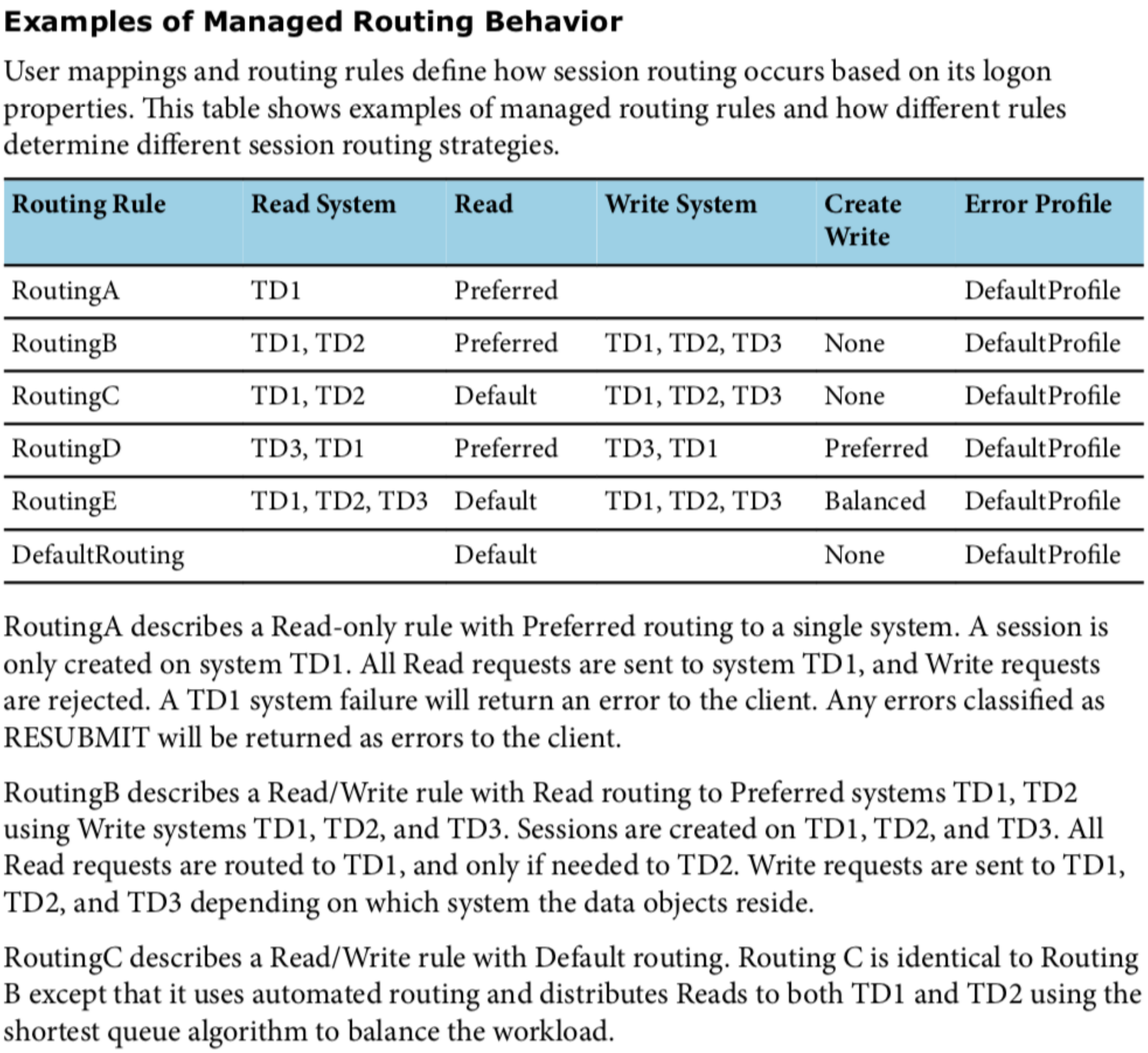
For Write requests, Unity does not wait for all responses from each Teradata Database system. Unity checks the first response for a successful operation or error code such as communication failures, memory errors, and disk space and database errors. If the first response is successful, notification is sent to the client application. If the first response is an error, all Active Teradata Database systems must return a response before the client application is notified.

The Active Sequencer determines which response is the most common successful response, and sends that response back to the client application. If no successful response is available, the most common error response is sent to Unity. If responses are at the same time, the first response that Unity receives returns to the client application. If the error code specifies the action, Exit, Unity returns the error and exits the session on all Teradata systems. If the response indicates a successful operation, Unity compares the results returned from all Teradata systems to determine consistency across the systems. All responses returned by the Teradata Database systems must match to verify data consistency.

Unity uses the activity count (number of rows) to indicate data consistency for all INSERT, DELETE, UPDATE, and MERGE statements. The activity count from all Teradata Database systems must match the activity count of the response sent back to the client. The activity count is not checked for any other types of Writes. For these type of statements, Unity only checks for statement success or failure. If the activity count or success/ failure status of the Write operation does not match, Unity automatically changes the table state to Interrupted or Unrecoverable depending on the type of statement.

37. The Accused instrumentalities include a communication apparatus comprising a controller configured to "transmit data in the communication apparatus based on result of the comparison and the determination." For example, the Accused Instrumentalities transmits data to the Teradata Database server to an external database system that is identified, when it is connected by a unique identifier called a TDPID, and the Accused Instrumentalities controls what data to transmit based on, e.g., managed routing rules, and edits management information based on the selection. See, e.g., 2520-066K.pdf ("DICTIONARY SET PREFERRED: Purpose: Sets the Preferred Write system to the Teradata Database system identified by the TDPID value. To set the Preferred Write system for all tables in a database, specify only the name of the database. To set the Preferred Write system for a single table, specify both the database name and the table name. *** tdpid: Unique identifier (TDPID) of a Teradata Database system."); EB7192.pdf ("Taken together, the data loading and query management work as one to sequence database changes and query requests. This ensures that data updates and database structure changes are always applied in the order received; automatically coordinating and maintaining order and consistency across systems. And, it is specifically designed to work

with systems that are not identical. For example, if the primary integrated data warehouse contains 100 percent of the data, and there is a second system for high availability that holds a 30 percent subset of that data, Teradata Unity understands the capabilities and limitations of each system and will route data loads accordingly."); *id.*:



Examples of Managed Routing Behavior

User mappings and routing rules define how session routing occurs based on its logon properties. This table shows examples of managed routing rules and how different rules determine different session routing strategies.

Routing Rule	Read System	Read	Write System	Create Write	Error Profile
RoutingA	TD1	Preferred			DefaultProfile
RoutingB	TD1, TD2	Preferred	TD1, TD2, TD3	None	DefaultProfile
RoutingC	TD1, TD2	Default	TD1, TD2, TD3	None	DefaultProfile
RoutingD	TD3, TD1	Preferred	TD3, TD1	Preferred	DefaultProfile
RoutingE	TD1, TD2, TD3	Default	TD1, TD2, TD3	Balanced	DefaultProfile
DefaultRouting		Default		None	DefaultProfile

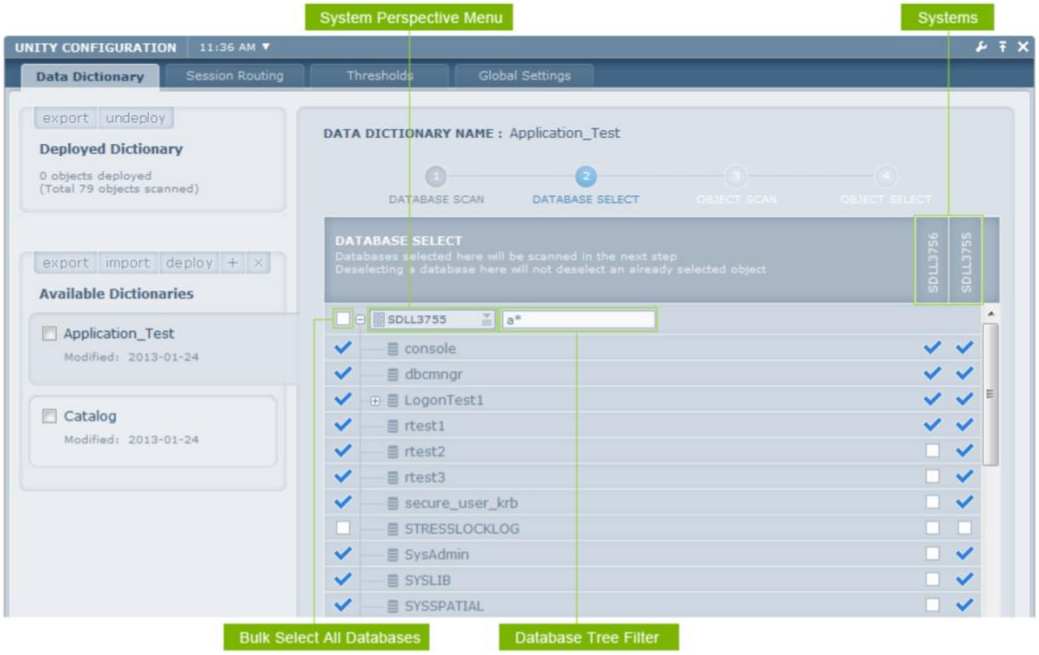
RoutingA describes a Read-only rule with Preferred routing to a single system. A session is only created on system TD1. All Read requests are sent to system TD1, and Write requests are rejected. A TD1 system failure will return an error to the client. Any errors classified as RESUBMIT will be returned as errors to the client.

RoutingB describes a Read/Write rule with Read routing to Preferred systems TD1, TD2 using Write systems TD1, TD2, and TD3. Sessions are created on TD1, TD2, and TD3. All Read requests are routed to TD1, and only if needed to TD2. Write requests are sent to TD1, TD2, and TD3 depending on which system the data objects reside.

RoutingC describes a Read/Write rule with Default routing. Routing C is identical to Routing B except that it uses automated routing and distributes Reads to both TD1 and TD2 using the shortest queue algorithm to balance the workload.

See also http://downloads.teradata.com/uda/articles/unity-director-initial-configuration-and-setup:

b. Select databases used in the application:



Deployed Dictionary

0 objects deployed
(Total 79 objects scanned)

Available Dictionaries

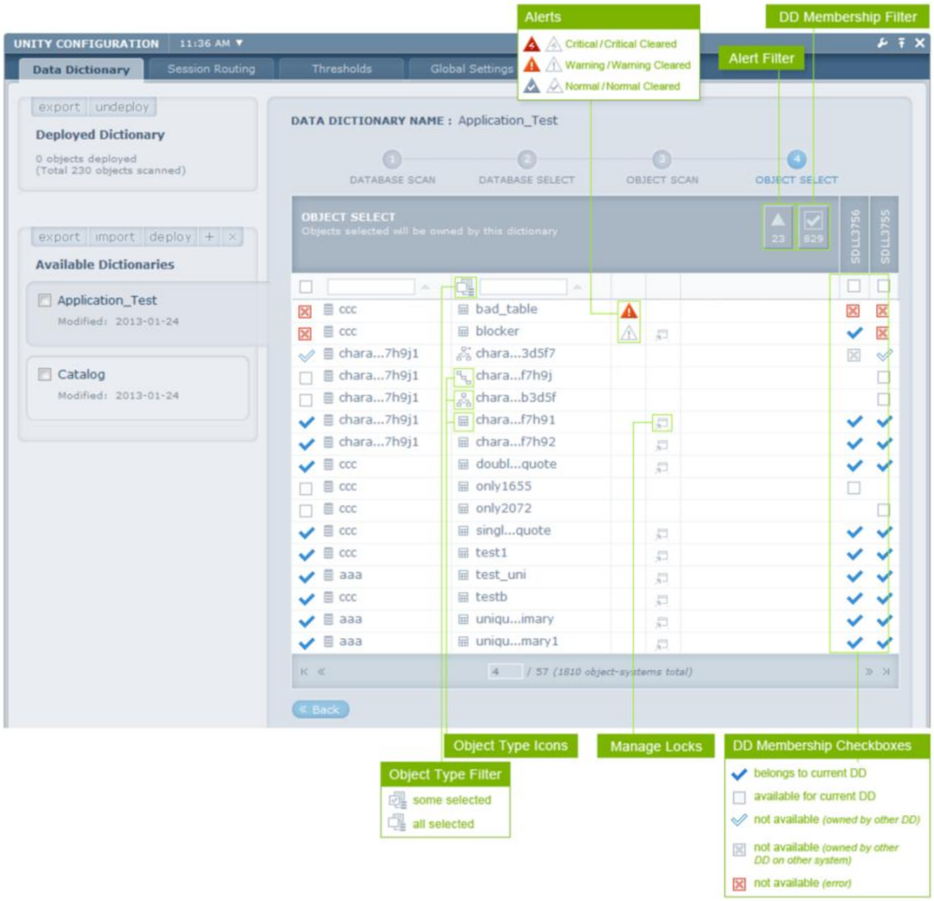
□ Application_Test
Modified: 2013-01-24

□ Catalog
Modified: 2013-01-24

DATA DICTIONARY NAME : Application_Test

DATABASE SELECT
Databases selected here will be scanned in the next step
Deselecting a database here will not deselect an already selected object

□	SDLL3755	a*	SDLL3756	SDLL3755
✓	console		✓	✓
✓	dbcmngr		✓	✓
✓	LogonTest1		✓	✓
✓	rtest1		✓	✓
✓	rtest2		□	✓
✓	rtest3		□	✓
✓	secure_user_krb		□	✓
□	STRESSLOCKLOG		□	□
✓	SysAdmin		□	✓
✓	SYSLIB		□	✓
✓	SYSPATIAL		□	✓



▲ △ Critical / Critical Cleared
▲ △ Warning / Warning Cleared
▲ △ Normal / Normal Cleared

Deployed Dictionary

0 objects deployed
(Total 230 objects scanned)

Available Dictionaries

□ Application_Test
Modified: 2013-01-24

□ Catalog
Modified: 2013-01-24

DATA DICTIONARY NAME : Application_Test

OBJECT SELECT
Objects selected will be owned by this dictionary

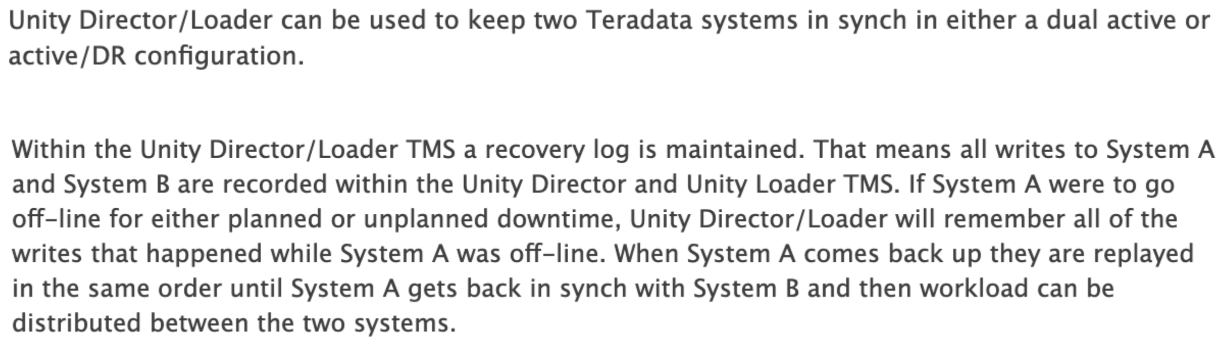
□					SDLL3756	SDLL3755
☒	ccc	bad_table	▲		☒	☒
☒	ccc	blocker	△	🔒	✓	☒
✓	chara...7h9j1	chara...3d5f7			☒	✓
□	chara...7h9j1	chara...f7h9j				□
□	chara...7h9j1	chara...b3d5f				□
✓	chara...7h9j1	chara...f7h91		🔒	✓	✓
✓	chara...7h9j1	chara...f7h92		🔒	✓	✓
✓	ccc	doubl...quote		🔒	✓	✓
□	ccc	only1655			□	
□	ccc	only2072				□
✓	ccc	singl...quote		🔒	✓	✓
✓	ccc	test1		🔒	✓	✓
✓	aaa	test_uni		🔒	✓	✓
✓	ccc	testb		🔒	✓	✓
✓	aaa	uniqu...imary		🔒	✓	✓
✓	aaa	uniqu...mary1		🔒	✓	✓

4 / 57 (1810 object-systems total)

Object Type Filter
- some selected
- all selected

DD Membership Checkboxes
- ✓ belongs to current DD
- □ available for current DD
- ✓ not available *(owned by other DD)*
- ☒ not available *(owned by other DD on other system)*
- ☒ not available *(error)*

See also, e.g., https://developer.teradata.com/general/articles/multi-active-systems-with-new-unity-directorloader-14-11:



Unity Director/Loader can be used to keep two Teradata systems in synch in either a dual active or active/DR configuration.

Within the Unity Director/Loader TMS a recovery log is maintained. That means all writes to System A and System B are recorded within the Unity Director and Unity Loader TMS. If System A were to go off-line for either planned or unplanned downtime, Unity Director/Loader will remember all of the writes that happened while System A was off-line. When System A comes back up they are replayed in the same order until System A gets back in synch with System B and then workload can be distributed between the two systems.

38. Defendant has had knowledge of the ’751 Patent and its infringement since at least the filing of the original Complaint in this action, or shortly thereafter, including by way of this lawsuit. By the time of trial, Defendant will have known and intended (since receiving such notice) that its continued actions would actively induce and contribute to the infringement of the claims of the ’ 751 Patent.

39. Defendant’s affirmative acts of making, using, selling, offering for sale, and/or importing the Accused Instrumentalities have induced and continue to induce users of the Accused Instrumentalities to use the Accused Instrumentalities in their normal and customary way to infringe the claims of the ’751 Patent. Use of the Accused Instrumentalities in their ordinary and customary fashion results in infringement of the claims of the ’751 Patent.

40. For example, Defendant explains to customers the benefits of using the Accused Instrumentalities, such as by touting their advantages of synchronizing settings among multiple devices. Defendant also induces its customers to use the Accused Instrumentalities to infringe other claims of the ’ 751 Patent. Defendant specifically intended and was aware that the normal and customary use of the Accused Instrumentalities

on compatible systems would infringe the ' 751 Patent. Defendant performed the acts that constitute induced infringement, and would induce actual infringement, with the knowledge of the '751 Patent and with the knowledge, or willful blindness to the probability, that the induced acts would constitute infringement. On information and belief, Defendant engaged in such inducement to promote the sales of the Accused Instrumentalities, e.g., through its user manuals, product support, marketing materials, demonstrations, installation support, and training materials to actively induce the users of the accused products to infringe the ' 751 Patent. Accordingly, Defendant has induced and continues to induce end users of the accused products to use the accused products in their ordinary and customary way with compatible systems to make and/or use systems infringing the ' 751 Patent, knowing that such use of the Accused Instrumentalities with compatible systems will result in infringement of the ' 751 Patent. Accordingly, Defendant has been (since at least as of filing of the original complaint), and currently is, inducing infringement of the ' 751 Patent, in violation of 35 U.S.C. § 271(b).

41. For similar reasons, Defendant also infringes the '751 Patent by supplying or causing to be supplied in or from the United States all or a substantial portion of the components of the Accused Instrumentalities, where such components are uncombined in whole or in part, in such manner as to actively induce the combination of such components outside of the United States in a manner that would infringe the '751 Patent if such combination occurred within the United States. For example, Defendant supplies or causes to be supplied in or from the United States all or a substantial portion of the hardware (e.g., separate Teradata servers) and software (e.g., Teradata Unity software) components of the Accused Instrumentalities in such a manner as to actively induce the combination of such

components (e.g., by instructing users to combine multiple Teradata servers into an infringing system) outside of the United States.

42. Defendant has also infringed, and continues to infringe, claims of the ’751 Patent by offering to commercially distribute, commercially distributing, making, and/or importing the Accused Instrumentalities, which are used in practicing the process, or using the systems, of the ’751 Patent, and constitute a material part of the invention. Defendant knows the components in the Accused Instrumentalities to be especially made or especially adapted for use in infringement of the ’751Patent, not a staple article, and not a commodity of commerce suitable for substantial noninfringing use. For example, the ordinary way of using the Accused Instrumentalities infringes the patent claims, and as such, is especially adapted for use in infringement. Accordingly, Defendant has been, and currently is, contributorily infringing the ’751 Patent, in violation of 35 U.S.C. § 271(c).

43. Defendants also indirectly infringe the ’751 Patent by supplying or causing to be supplied in or from the United States components of the Accused Instrumentalities that are especially made or especially adapted for use in infringing the ’ 751 Patent and are not a staple article or commodity of commerce suitable for substantial non-infringing use, and where such components are uncombined in whole or in part, knowing that such components are so made or adapted and intending that such components are combined outside of the United States in a manner that would infringe the ’751 Patent if such combination occurred within the United States. Because the Accused Instrumentalities are designed to operate as the claimed system and apparatus, the Accused Instrumentalities have no substantial non-infringing uses, and any other uses would be unusual, far-fetched, illusory, impractical, occasional, aberrant, or experimental. For example, Defendant

supplies or causes to be supplied in or from the United States all or a substantial portion of the hardware (e.g., separate Teradata servers) and software (e.g., Teradata Unity software) components that are especially made or especially adapted for use in the Accused Instrumentalities, where such hardware and software components are not staple articles or commodities of commerce suitable for substantial noninfringing use, knowing that such components are so made or adapted and intending that such components are combined outside of the United States, as evidenced by Defendant's own actions or instructions to users in, e.g., combining multiple Teradata servers into infringing systems, and enabling and configuring the infringing functionalities of the Accused Instrumentalities.

44. As a result of Defendant's infringement of the '751 Patent, Plaintiff Data Scape is entitled to monetary damages in an amount adequate to compensate for each Defendant's infringement, but in no event less than a reasonable royalty for the use made of the invention by each Defendant, together with interest and costs as fixed by the Court.

<u>PRAYER FOR RELIEF</u>

WHEREFORE, Plaintiff Data Scape respectfully requests that this Court enter:

a. A judgment in favor of Plaintiff that Defendant has infringed, either literally and/or under the doctrine of equivalents, the '675 Patent and the '751 Patent (collectively, "asserted patents");

b. A permanent injunction prohibiting Defendant from further acts of infringement of the asserted patents;

c. A judgment and order requiring Defendant to pay Plaintiff its damages, costs, expenses, and prejudgment and post-judgment interest for its infringement of the asserted patents, as provided under 35 U.S.C. § 284;

d. A judgment and order requiring Defendant to provide an accounting and to pay supplemental damages to Data Scape, including without limitation, prejudgment and post-judgment interest;

e. A judgment and order finding that this is an exceptional case within the meaning of 35 U.S.C. § 285 and awarding to Plaintiff its reasonable attorneys' fees against Defendant; and

f. Any and all other relief as the Court may deem appropriate and just under the circumstances.

DEMAND FOR JURY TRIAL

Plaintiff, under Rule 38 of the Federal Rules of Civil Procedure, requests a trial by jury of any issues so triable by right.

Dated: May 20, 2019

Respectfully submitted,

/s/ Marc A. Fenster
Marc A. Fenster

Marc A. Fenster (CA SBN 181067)
Email: mfenster@raklaw.com
Reza Mirzaie (CA SBN 246953)
Email: rmirzaie@raklaw.com
Brian D. Ledahl (CA SBN 186579)
Email: bledahl@raklaw.com
Paul A. Kroeger (CA SBN 229074)
Email: pkroeger@raklaw.com
C. Jay Chung (CA SBN 252794)
Email: jchung@raklaw.com
Philip X. Wang (CA SBN 262239)
Email: pwang@raklaw.com

RUSS AUGUST & KABAT
12424 Wilshire Blvd., 12th Floor
Los Angeles, California 90025
Telephone: (310) 826-7474
Facsimile: (310) 826-6991

Attorneys for Plaintiff
Data Scape Limited